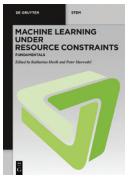
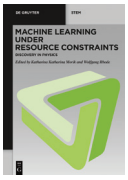
Katharina Morik, Jörg Rahnenführer, Christian Wietfeld (Eds.)
Machine Learning under Resource Constraints · Applications

Also of interest



Volume 1
Machine Learning under Resource Constraints. Fundamentals
Morik, Marwedel (Eds.), 2023
ISBN 978-3-11-078593-7, e-ISBN 978-3-11-078594-4



Volume 2
Machine Learning under Resource Constraints. Discovery in Physics
Morik, Rohde (Eds.), 2023
ISBN 978-3-11-078595-1, e-ISBN 978-3-11-078596-8

Machine Learning under Resource Constraints

Final Report of CRC 876

Editor in Chief
Katharina Morik

Volume 3/3

DE GRUYTER

Machine Learning under Resource Constraints

Applications

Edited by
Katharina Morik, Jörg Rahnenführer and
Christian Wietfeld

DE GRUYTER

Editors
Prof. Dr. Katharina Morik
TU Dortmund University
Department of Computer Sciences
Chair for Artificial Intelligence
Computer Science 8
Otto-Hahn-Str. 12
44221 Dortmund
Germany

Prof. Dr. Christian Wietfeld
TU Dortmund University
Department of Electrical Engineering and
Information Technology
Chair for Communication Networks
Otto-Hahn-Straße 6
44221 Dortmund
Germany

Prof. Dr. Jörg Rahnenführer
TU Dortmund University
Department of Statistics
Statistical Methods in Genetics and
Chemometrics
Vogelpothsweg 87
44227 Dortmund
Germany

ISBN 978-3-11-078597-5
e-ISBN (PDF) 978-3-11-078598-2
e-ISBN (EPUB) 978-3-11-078614-9
DOI https://doi.org/10.1515/9783110785982



This work is licensed under the Creative Commons Attribution 4.0 International License. For details go to https://creativecommons.org/licenses/by/4.0/.

Creative Commons license terms for re-use do not apply to any content (such as graphs, figures, photos, excerpts, etc.) not original to the Open Access publication and further permission may be required from the rights holder. The obligation to research and clear permission lies solely with the party re-using the material.

Library of Congress Control Number: 2022949226

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

© 2023 with the author(s), editing © 2023 Katharina Morik, Jörg Rahnenführer and Christian Wietfeld, published by Walter de Gruyter GmbH, Berlin/Boston
This book is published open access at www.degruyter.com.

Cover image: Collaborative Research Center 876
Printing and binding: CPI books GmbH, Leck

www.degruyter.com

Contents

1 Editorial — 1

2 Health / Medicine — 3

2.1 **Machine Learning in Medicine**
Catherine Jutzeler, Karsten Borgwardt — **3**

2.2 **Virus Detection**
Roland Hergenröder, Frank Weichert, Konstantin Wüstefeld, Victoria Shpacovitch — **21**

2.3 **Cancer Diagnostics and Therapy from Molecular Data**
Sven Rahmann, Alexander Schramm, Johannes Köster — **43**

2.4 **Bayesian Analysis for Dimensionality and Complexity Reduction**
Zeyu Ding, Katja Ickstadt, Alexander Munteanu — **58**

2.5 **Survival Prediction and Model Selection**
Jörg Rahnenführer, Michel Lang, Jakob Richter — **71**

2.6 **Protein Complex Similarity**
Bianca K Stöcker, Till Schäfer, Petra Mutzel, Johannes Köster, Nils Kriege, Sven Rahmann — **85**

3 Industry 4.0 — 103

3.1 **Keynote on Industry 4.0**
Michael ten Hompel, Moritz Roidl — **103**

3.2 **Quality Assurance in Interlinked Manufacturing Processes**
Jochen Deuse, Katharina Morik, Amal Saadallah, Jan Büscher, Thorben Panusch — **114**

3.3 **Label Proportion Learning**
Marco Stolpe, Katharina Morik — **136**

3.4 **Simulation and Machine Learning**
Petra Wiederkehr, Katharina Morik, Amal Saadallah, Felix Finkeldey — **157**

3.5 **High-Precision Wireless Localization**
Janis Tiemann — **180**

3.6 **Indoor Photovoltaic Energy Harvesting**
Mojtaba Masoudinejad — **195**

3.7 **Micro-UAV Swarm Testbed for Indoor Applications**
Nils Gramse, Moritz Roidl, Shrutarv Awasthi, Christopher Reining — **212**

4 Smart City and Traffic — 225

4.1 **Inner-City Traffic Flow Prediction with Sparse Sensors**
Thomas Liebig — **225**

4.2 **Privacy-Preserving Detection of Persons and Classification of Vehicle Flows**
Marcus Haferkamp, Benjamin Sliwa, Christian Wietfeld — **241**

4.3 **Green Networking and Resource Constrained Clients for Smart Cities**
Pascal Jörke, Christian Wietfeld — **261**

4.4 **Vehicle to Vehicle Communications: Machine Learning-Enabled Predictive Routing**
Cedrik Krieger, Benjamin Sliwa, Christian Wietfeld — **272**

4.5 **Modelling of Hybrid Vehicular Traffic with Extended Cellular Automata**
Michael Schreckenberg, Tim Vranken — **285**

4.6 **Embedded Crowdsensing for Pavement Monitoring and its Incentive Mechanisms**
Maximillian Machado, Ran Ran, Liang Cheng — **297**

5 Communication Networks — 313

5.1 **Capacity Analysis of IoT Networks in Unlicensed Spectrum**
Stefan Böcker, Christian Arendt, Christian Wietfeld — **313**

5.2 **Resource-Efficient Vehicle-to-Cloud Communications**
Benjamin Sliwa — **331**

5.3 **Mobile-Data Network Analytics of Highly Reliable Networks**
Robert Falkenberg, Karsten Heimann, Benjamin Sliwa — **342**

5.4 **Machine Learning-Enabled 5G Network Slicing**
Caner Bektas, Dennis Overbeck, Fabian Kurtz, Christian Wietfeld — **354**

5.5 **Potential of Millimeter Wave Communications**
Karsten Heimann, Simon Häger, Christian Wietfeld — **375**

6 Privacy — 391

6.1 **Keynote: Construction of Inference-Proof Agent Interactions**
Joachim Biskup — **391**

Bibliography — 413

Index — 461

List of Contributors — 467

1 Editorial

Katharina Morik
Jörg Rahnenführer
Christian Wietfeld

This is the third book of a series of books dedicated to the results of the DFG Collaborative Research Center 876 on "Machine Learning under Resource Constraints". The first book of the series discusses fundamental innovations in the theory and algorithms of machine learning. The second book covers the use of machine learning in physics. This volume focuses on applications of the machine learning approaches presented in Book 1. The main idea is to demonstrate with specific examples how machine learning has become essential as well as practical in solving real-life problems from diverse application areas, ranging from medicine and robotics to road traffic and communication networks. Various real-life example applications show the significant impact of using tailored machine learning methods to improve the performance of the respective processes and systems. A key boundary condition imposed by the real-life environments is that resources, such as energy, storage, computing power, computing time, etc., are often limited and that the practicability of the proposed machine learning solutions depends on the efficient use of those resources. Therefore, the success of the solutions discussed in this book must not only be measured in terms of performance gains but, at the same time, in terms of their resource efficiency and corresponding sustainability. For many domain experts, the sheer multitude of machine learning approaches makes it difficult to choose the "right" ones for a particular problem. While the availability of software tools lowers the entry barrier to use machine learning methods by non-experts, the application examples contained in this book demonstrate that truly significant impacts can often only be achieved by an interdisciplinary combination of domain knowledge and the appropriate usage of machine learning methods. Accordingly, this book aims to promote proficiency in the use of machine learning methods beyond the quick wins of arbitrarily using whatever approach happens to be in fashion. The applications described in this book will touch upon a multitude of machine learning options covering the complete process chain from data acquisition, feature extraction, model selection via various learning approaches to model verification and model validation. It will show, for example, that while the deep learning approaches popular today can be beneficial for many problems in some areas, alternative methods such as ensemble learning with random forests are more accurate with much less resource utilization in other areas. The first part of the book addresses the application area of health and medicine. After an overview of machine learning in medicine provided by the invited authors Catherine Jutzeler and Karsten Borgwardt, a number of results from the CRC 876 are presented, covering virus detection, protein analysis, and cancer diagnostics and therapy. The

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.
https://doi.org/10.1515/9783110785982-001

second part of the book is dedicated to the application of machine learning for industry use cases. On the one hand, machine learning enables proactive quality assessments; on the other, its application for precise localization, energy harvesting, and swarm control demonstrates the potential of embedding machine learning methods in almost any element of the manufacturing and logistics process of present and future industry environments. In the third part of the book, various examples illustrate the significant potential of machine learning for smart city and traffic use cases, such as the prediction of traffic flows, the privacy-preserving detection of vehicle flows, and resource-efficient crowd sensing in smart cities.

The fourth part of the book is about improving the performance of communication networks through machine learning. This includes new approaches for highly resource-efficient vehicle-to-cloud communications as well as machine learning-enabled mobile data network analytics and proper dimensioning of 5G network slices. As many applications of machine learning involve personal data and may affect privacy concerns, this book also includes a chapter on a general methodology to classify and handle privacy aspects of data management as part of the machine learning process chain. This focus on the data handling complements the privacy-preserving machine learning techniques. With this broad spectrum of application and practical implementation examples, we hope that this book will serve domain experts from diverse application areas as inspiration for the use of machine learning for their application-specific problems. To maximize the impact, many of the presented solutions are provided as open source published together with open data sets, allowing for reproducibility and sustainable transfer. At the same time, machine learning experts are expected to be motivated by the impressive impact of their work on real-life problems to further expand the machine learning solution space in terms of accuracy and resource efficiency.

Dortmund, 14.10.2022
Katharina Morik, Jörg Rahnenführer and Christian Wietfeld

2 Health / Medicine

2.1 Machine Learning in Medicine

Catherine Jutzeler
Karsten Borgwardt

Abstract: The combination of machine learning and population-scale health data holds the potential to revolutionize disease diagnosis and prognosis as well as to enable personalized predictions of therapy responses. The foundation for this unique opportunity is the ever-increasing amount of complex high-dimensional health data, from the molecular level of genome sequences to the level of image phenotypes and health history, that is available in digital form and at high resolution for cohorts of hundreds of thousands—and soon millions—of individuals. Machine learning promises to be a key technology in the generation of new knowledge from this big health data, by detecting new statistical dependencies in large and multisource medical datasets. These new data-driven insights may drastically improve our abilities to predict disease onset early, define sub-types of diseases, and model disease progression and patient heterogeneity at an unprecedented level of detail, thereby supporting clinical decisions. Nevertheless, the practical implementation of the vision of machine learning-guided precision medicine faces considerable clinical challenges that have to be addressed in the future. In this contribution, we will describe the envisioned role of machine learning in the context of healthcare, critically discuss the challenges faced in terms of the implementation in the clinical routine, and outline future directions of this growing field.

2.1.1 Introduction: The Envisioned Role of Machine Learning in Precision Medicine

Precision medicine envisions that medical diagnosis, prognosis, and interventions can be tailored to the clinical, molecular, and genetic signature of individual patients [278]. One promising path towards precision medicine is to exploit the explosion of health data with modern computational approaches, in particular machine learning. Machine learning can be leveraged to detect hidden signals in digital health data (e.g., risk factors), uncover patterns or associations with certain diseases, and evaluate the outcome of treatments or interventions. Applications of machine learning have also been proposed to facilitate early disease recognition, refine diagnosis and prognosis, support therapy decisions, and improve biomedical data management. An ever-increasing amount of data, from the molecular level of genome sequences to the level of image

Open Access. © 2023 the author(s), published by De Gruyter. CC BY This work is licensed under the Creative Commons Attribution 4.0 International License.
https://doi.org/10.1515/9783110785982-002

phenotypes and health history, is available for rapidly growing cohorts of individuals. A prime example is the UK Biobank [658], which makes health data (genetic, molecular, imaging, clinical data) of more than half a million healthy people and patients available to the global research community. Exploring population-scale health data presents enormous opportunities for understanding disease mechanisms, ameliorating therapy outcomes, and ultimately improving healthcare. Machine learning and artificial intelligence offer the methods to mine and analyse the vast amounts of high-dimensional digital health data. For instance, designing computational models of diseases opens new opportunities to refine our understanding of diseases and their subtypes, discover novel biomarkers for early disease detection, and guide clinical decisions. A crucial step towards realizing the vision of precision and eventually personalized medicine, will be the ability of machine learning algorithms to simultaneously compute and consider a multitude of patient characteristics. The problem is exacerbated by the fact that current machine learning applications are often restricted by (1) a lack of patient data, let alone patient data with temporally-resolved clinical phenotypes; by (2) massive missingness in longitudinal, multi-modal patient data; by (3) the enormous effort required to combine data from different hospitals, with legal, information technology (IT), and data engineering challenges.

The remainder of this contribution will provide an overview of machine learning applications in the field of medicine, with a special emphasis on early disease recognition, diagnosis, prognosis, and therapy decisions. Moreover, we will critically discuss the challenges of machine learning-guided applications in the context of medicine and health care in general. Lastly, we conclude with an outlook of what the future may hold for machine learning-driven applications in the different areas of health care, such as diagnosis, prognosis, and therapy development.

2.1.2 Overview of Common Topics in Machine Learning in Medicine

The notion of advancing medicine by means of computation is almost as old as digital computers [438]. When deployed into the clinical routine, machine learning-guided approaches can facilitate early disease recognition, refine diagnosis and prognosis, support therapy decisions, and improve biomedical data management [161]. In this section, we discuss studies that illustrate the potential role of machine learning to tackle these tasks.

2.1.3 Automation of Diagnoses and Treatment

Depending on the disease type, time is often a limiting factor for the diagnosis and initiation of effective treatment. This is particularly true for patients facing serious medical conditions (e.g., cardiovascular complications, sepsis, cancer), which require

immediate attention and timely clinical decisions. A delay in the diagnostic work-up puts the patient at risk because the medical condition can get worse the longer it remains undiagnosed and untreated. To accelerate the diagnostic workup the medical field has increasingly used automation in diagnostics, surgical planning, and therapy selection (Table 2.1). Nowadays, machine learning models play an important role in the development and implementation of automation processes in the clinical routine. Large clinical datasets provide ample amounts of 'raw' data from which machine learning algorithms can derive clinically relevant insights that can inform the diagnosis or treatment selection of a patient. Specific examples, which will be discussed in detail, are the timely identification of circulatory failure [293], automatic antimicrobial resistance prediction [722], cancer tumor recognition in radiology images [28, 123], and automation of treatment planning in oncology [713].

A prominent example for automated diagnosis is circulatory failure, which occurs when the arterial pressure and capillary stream are reduced for a prolonged period of time [89]. Subsequently, the functions of supplied organs are impaired or in the worst case even lost. As circulatory failure is common in critically ill patients, monitoring of circulatory function is an indispensable aspect of the patient management in the Intensive Care Unit (ICU). Short-term effects of circulatory failure are usually reversible, whereas repeated or extended episodes of low arterial pressure adversely affect outcomes and worsen the prognosis [184, 537]. Therefore, the early recognition of circulatory failure is of highest priority. Conventional alarm systems to detect circulatory failure do not utilize comprehensive patient information, which often lead to alarms that are non-specific or false [553, 619]. False or unspecific alarms can trigger "alarm fatigue" among intensive care practitioners [97]. In the ICU, large quantities of measurements from multiple monitoring systems are generated that carry clinically relevant information. While too complex to analyze for a human brain, machine learning applications thrive in data-rich environments, such as the ICU. Leveraging clinical and ICU monitoring data, Hyland and colleagues show that a machine learning algorithm based on an array of demographic, physiological, and clinical information is able to predict the circulatory failure of ICU patients several hours prior to its onset [293]. Their early-warning system outperforms current conventional threshold-based systems and has a significantly lower false-alarm rate. In order to learn to detect deterioration events from monitoring data, which are indicative of circulatory failure, different state-of-the-art supervised machine learning techniques were employed, including Light Gradient Boosting Machine (lightGBM) [315], Logistic Regression [283], and Long Short-Term Memory (LSTM) based recurrent neural network model [277]. When implemented in the clinic, such machine learning guided multi-modal early-warning systems are a first step towards (semi-) automation of the identification of patients at risk for the development of circulatory failure in the ICU, while avoiding "alarm fatigue" among the clinical staff.

In addition to the early recognition of circulatory failure or other serious conditions (e.g., sepsis), a major challenge faced by intensive care practitioners is the escalating rates of antibiotic resistance in ICU patients. The administration of antibiotics is up to

Tab. 2.1: Selected examples of proposed machine learning approaches that could guide the automation of early detection, disease diagnosis, and treatment planning.

Disease	Automated task	Input data	Analytical methods used
Antibiotic resistance [722]	Diagnostic and treatment support	MALDI-TOF mass spectra and antimicrobial resistance profiles, demographics	Logistic Regression, Light Gradient Boosting Machine, Multilayer Perceptron Deep Neural Network
Circulatory failure [293]	Early detection	Physiological parameters, blood values, vitals, monitoring data, demographics	Light Gradient Boosting Machine, Logistic Regression, and Long Short-Term Memory based Recurrent Neural Network model
COVID-19 [487]	Detection	X-ray images, demographics, clinical data	Deep Learning, Convolutional Neural Network
Diabetic retinopathy [364]	Early detection	Fundoscope images, diabetic retinopathy images	Deep Learning, Convolutional Neural Network
Hyperlipidemia [760]	Diagnosis	Blood parameters, urine parameters, biochemical test parameters, blood sugar parameters, and glycosylated hemoglobin parameters	1-D Convolutional Neural Network
Laparoscopic robotic surgery [35]	Surgical path planning	Gall bladder images	Reinforcement Learning
Multiple sclerosis [659]	Detection	Brain MRI images, clinical data	Deep Learning, Convolutional Neural Network
Plastic and reconstructive surgery [329]	Diagnosis and surgical planning	3D face surface scans, demographics	Linear Regression, Ridge Regression, Least-Angle Regression, and Least Absolute Shrinkage and Selection Operator Regression, Support Vector Machine
Prostate cancer [569]	Therapy selection	Prostate cancer images, clinical data	Deep Learning Neural Networks
Prostate cancer [464]	Therapy selection, dose optimization	Prostate cancer computed tomography images, clinical data	Generative Adversarial Network

ten times higher in ICU patients compared with non-ICU patients [542]. Moreover, the proportion of antimicrobial resistant isolates was found to be considerably higher in ICU patients than in patients on general medical floors [27]. Antibiotic resistance substantially adds to the morbidity, mortality, and healthcare cost related to infections in the ICU [118]. Timely initiation of effective antimicrobial treatment has emerged as a critical determinant of outcome in patients with bacterial infections. The selection of optimal treatment warrants an exact characterization of the underlying pathogen, including the determining of resistance profiles. As time is of the essence, streamlining the antimicrobial resistance profiling is imperative. Matrix-Assisted Laser Desorption/Ionization Time-of-Flight (MALDI-TOF) mass spectrometry (MS) has become the standard rapid technology for the identification of microbial species, at least in specialised centers. Multiple studies have suggested that machine learning algorithms could be used to thoroughly exploit the information contained in MALDI-TOF MS spectra in order to expedite species identification and antimicrobial resistance determination [723]. However, there is a lack of comprehensive information on marker mass for all existing pathogens and antibiotics, impeding the interpretation of MALDI-TOF spectra. In a seminal study, Weis and colleagues used machine learning to harness the full potential of MALDI-TOF MS of microbial isolates to predict antimicrobial resistance [722]. Both of the implemented machine learning algorithms, Light Gradient Boosting Machine and a multilayer perceptron deep neural network, could reliably identify antimicrobial resistance of clinically important pathogens, including *Staphylococcus aureus* (*S. aureus*), *Escherichia coli* (*E. coli*), and *Klebsiella pneumoniae* (*K. pneumoniae*). Moreover, high predictive performance was observed for individual species–antibiotic combinations, such as ceftriaxone resistance in *E. coli* and *K. pneumoniae* and oxacillin resistance in *S. aureus* [722]. A retrospective clinical trial further demonstrated the clinical benefit of machine learning guided antibiotic resistance profiling. Based on the results provided by machine learning algorithms, the empiric antibiotic regimes would have changed for a subset of patients (≈ 15 %). Importantly, such a change would have been beneficial for the majority of patients. This study constitutes the first step of automatic phenotype determination, which promises to accelerate the diagnostic work up and guide treatment selection. Consequently, this will reduce the time from diagnosis to initiation of antibiotic treatment for ICU patients.

Another medical condition that will likely benefit from machine learning guided automated diagnosis is lung cancer, which is the world's the leading cause of cancer-related deaths [45]. Despite recent advances in the treatment of lung cancer, the overall 5-year survival is still low for advanced stages with distant metastases [616]. Initial symptoms of lung cancer tend to be unspecific (e.g., coughing, fatigue) and thus, are easy to dismiss as inconsequential. Consequently, the majority of patients present at an advanced disease stage when curative treatment is out of reach. Early diagnosis of lung cancer is thus imperative to increase the likelihood of survival and treatment success [45]. A successful strategy to significantly reduce the mortality is the regular screening of at-risk populations [338]. Yet, up to one third of lung nodules are missed

at the initial screening, likely owing to low the sensitivity and specificity of current screening methods as well as the limits of human vision. Imaging (e.g., X-ray, computer tomography [CT], or positron emission tomography-computerized tomography [PET-CT]) is an integral part of the diagnostic workup for lung cancer. Evaluation of the images is based on a number of imaging attributes, including the nodule size, density, and growth [511]. The detection of informative imaging features is a prime example of an area in which machine learning and artificial intelligence can excel and be of great value to the clinicians in terms of precision and time effectiveness. In particular, approaches based on deep learning [236], a branch of artificial intelligence, are an intriguing option for automating the complex image analysis to detect subtle alterations that specialists might overlook. In a seminal study, Ardila and colleagues developed three-dimensional Convolutional Neural Network (CNN) models that perform end-to-end analysis of CT images of pathology-confirmed lung cancer images [28]. Importantly, the model learns the features of interest as opposed to previous models that use hand-engineered features. Learning features have been repeatedly shown to be superior to hand-engineered features [415, 554]. The developed model was demonstrated to generate highly accurate patient-level malignancy risk predictions, which has important potential for clinical relevance. If clinically validated, the results of this study may represent a step toward automated image evaluation for risk malignancy estimation by means of deep learning. Importantly, though deep learning systems might outperform human specialists on some diagnostic tasks, they will not replace the radiologists but provide diagnostic guidance. When making a clinical decision, clinicians take into account a variety of factors that are not necessarily captured in the input data used by the machine learning model.

Along with early disease detection and phenotype detection, machine learning based automation has been demonstrated to be useful in the context of treatment planning. One striking example is Automated Therapy Planning (ATP) in patients with cancer that require radiotherapy [713]. The treatment success is highly dependent on the quality of the treatment plan. Inverse planning, a trial-and-error iterative process [512], is conventionally used to tailor radiotherapy treatment to the individual patients–a strategy that is strenuous and burdensome for patients. Machine learning and deep learning have gained momentum in the field as they are thought to improve the quality and efficiency of radiotherapy treatment planning. Specifically, the learning capability of these techniques enable oncologists to tailor the treatment plans to individual patients based on patient-specific anatomical features and by incorporating knowledge from optimization methods or physicians' behaviors. A variety of machine learning and deep learning techniques, from Artificial Neural Networks (ANN) [716], Convolutional Neural Networks (CNNs) [480], to Generative Adversarial Networks (GANs) [236], have been explored and incorporated in the different stages of radiotherapy treatment planning [44, 395, 570, 603]. While these methods promise to refine the therapy planning of various types of cancer, there are some issues relating to patient safety as well as legal

and ethical responsibilities that have to be considered before deep learning-based ATP can be implemented in the clinical routine.

2.1.4 Biomarker Discovery

Biomarker discovery, the search for measurable and reproducible indicators of specific clinical states, has been a major research avenue in recent years. A biomarker constitutes a measurable and reproducible indicator of specific clinical state or response to an intervention. In addition to refining disease diagnosis and prognosis [103, 139, 286], biomarkers of all sorts (e.g., molecular, digital, imaging) are instrumental in discovering and defining therapeutic targets [211, 601]. Data from various sources, including electronic health records, patient monitoring, and imaging, can be leveraged for biomarker discovery. With the emergence of affordable and time-efficient high-throughput molecular and gene expression profiling technologies (e.g., DNA microarrays and RNA sequencing) [282], the search space for biomarkers has reached unprecedented dimensions and complexity. The challenging nature of these datasets (e.g., high dimensionality with large number of noisy features and low sample size) require suitable computational models that thrive in these complex, data-rich environments. In light of that, a variety of machine learning-guided biomarker discovery strategies have gained great popularity across different fields of medicine [112, 362, 456, 667] (Table 2.2).

At the core of biomarker discovery is the search for markers that can discriminate between samples or clinical characteristics of diseased patients and those of healthy subjects. Biomarker discovery is equivalent to feature selection in machine learning [595]. Feature selection algorithms are intended to reduced the dimensionality of the feature space by removing non-informative and redundant features, while retaining the informative features [595]. In general, feature selection algorithms can be used to (i) modify representations of data by extracting informative variables (i.e., feature extraction), (ii) create probabilistic models of disease progression, and (iii) determine what specific piece of (unknown) information, for instance laboratory tests, will be most valuable in optimizing the predictive ability of a model. Broadly speaking, there are two major strategies of feature selection. The first is univariate feature selection that investigates each feature one by one to determine the strength of the relationship with the outcome variable. Popular univariate feature selection methods include linear mixed models [416], support vector machine [764], and kernel-based measures [142]. Variants of these models tackle the challenging problem of feature selection from time series data [91, 92, 320] and can account for covariates or confounders [23, 419]. The second major strategy is multivariate feature selection, which considers whole groups of features together. Common multivariate prediction models are Lasso regression models , tree ensembles (i.e., gradient boosting trees) [315], support vector machines, and Gaussian processes with kernels for comparing time series [430], neural networks from

Tab. 2.2: Selected examples of machine learning applications in the context of biomarker discovery.

Disease	Use of biomarker	Input data	Analytical methods used
Alzheimer's disease [456]	Disease progression	MRI brain images, clinical data	Logistic Regression, Support Vector Machine
Autism [173]	Disease diagnosis	Functional connectivity, structural connectivity, behavioral data, and brain activation measures	Recursive Cluster Elimination based Support Vector Machine
COVID-19 [428]	Mortality prediction	Laboratory values, demographics, medical history	Support Vector Machine
COVID-19 [590]	Mortality prediction	Laboratory data, clinical data, demographics	Cox Proportional Hazard Model
Dermatitis [211]	Disease type discrimination	Transcriptomics, skin biopsies, clinical data	Multi-island Adaptive Genetic Algorithm, Principle Component Analysis
Diabetes [261]	Prediction of disease progression	Physiological, biochemical, and sequencing data	Decision Trees, Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors Classifier, Gaussian Naïve Bayes, and Support Vector Machine
Huntington's disease [538]	Early detection	Structural and functional MRI, diffusion weighted-imaging scans	Linear Discriminant Analysis, Support Vector Machine
Lung cancer [734]	Early detection	Plasma metabolomic data	AdaBoost, K-nearest neighbor, Naïve Bayes, Neural Network, Random Forest, Support Vector Machine
Prostate cancer [284]	Screening and diagnosis	Microarray data, cancer tissue	Artificial neural network
Prostate cancer [139]	Diagnosis and prognosis	Proteomic data	Random Forest, Brute Force
Sepsis [454]	Early detection	Physiological parameters, blood values, vitals, monitoring data, demographics	Gaussian Process Temporal Convolutional Networks and Dynamic Time Warping
Traumatic brain injury [448]	Diagnosis	Structural MRI data, clinical data, demographics	Principle Component Analysis, Random Forest
Tuberculosis [363]	Disease detection	Chest X-Ray images, radiology reports, clinical data	Convolutional Neural Networks

deep learning (long short-term memory [277], gated recurrent units [135], temporal convolutional networks [38], and attention models [702]. Several of these techniques have been successfully applied to clinical outcome prediction over recent years, especially in the area of intensive care research [216, 293, 454, 455].

The ongoing coronavirus disease (COVID-19) pandemic has been a shining example of how machine learning can support and even guide the biomarker discovery. At the beginning of the pandemic, there were no recommendations or guidelines in place on how to manage COVID-19 patients or identify patients at risk. As a consequence, physicians in emergency and intensive care units were deemed to improvise on an individual patient level and administer off-label treatments. This was particularly difficult for patients who appeared to be on a disease trajectory towards recovery, but suddenly deteriorate at a speed that does not allow for timely targeted management. COVID-19 has been associated with a high 'failure-to-rescue' rate (i.e., number of deaths of a patient following a complication) [618]. Further complicating COVID-19 disease management was the limited understanding of the different clinical phenotypes associated with COVID-19 [672] and how forthcoming mutations of SARS-CoV-2 will modify the clinical manifestation and/or responses to current off-label treatments. In combatting the COVID-19 pandemic, massive global efforts have been undertaken to determine factors that are associated with the clinical presentation of the disease and its progression [88, 205, 575], in-hospital mortality risk [428, 742], and treatment response [396]. With the availability of COVID-19-related clinical, imaging, and multi-omics data, clinically relevant biomarker signatures can be determined by means of computational modeling [428, 430, 590]. A recent study leveraged electronic clinical trial data from 69 hospitals to develop a risk-scoring system for assessing COVID-19 related in-hospital mortality risk [590]. A wide range of biomarkers (age, pre-existing cardiovascular issues, blood markers) were found to be significantly associated with mortality outcomes.

In conclusion, machine learning-driven applications can be found across various medical disciplines and bear great potential to advance health care as a whole. Nevertheless, it is important to mention that there are many challenges (e.g., data privacy, quality of the data, generalizability of the models) that have to be tackled on the road to the clinical implementation.

2.1.5 Biomedical Data Management

As data collection and volume surges, machine learning has emerged as a key player in the data management, easing the burden of querying data source, as well as the curation and governance of data. In the context of healthcare, machine learning-guided data management ranges from genome assembly to managing national electronic health record systems. In general, data management is a labor- and time-intensive task that often involves repetitive steps, which can be (partially) automated by means of machine learning (Table 2.3). Machine learning algorithms pursue three major data management

goals: automation of time-consuming and iterative development tasks (cataloging data, mapping sources to targets, data preprocessing), optimization of system performance (table-join approaches), and capacity management (workload-aware autoscaling).

For instance, preparing and cleaning the raw data and making it suitable for subsequent analysis is an integral part of creating any statistical or machine learning model. Data preprocessing entails different steps: datasets requests, data fusion, and data anomaly detection. Defining the quality of the data is an important step as it will directly impact the performance and reliability of machine learning algorithms. Anomaly detection aims to identify observations or data elements that raise suspicions as they significantly deviate from the majority of the data. Anomalous data can be indicative of the data-quality issues, nonstandard data, or outliers. Machine learning algorithms have the potential to automatically detect and remediate data-quality issues [138]. Specific examples of machine learning applied to anomaly detection and data cleaning are clustering [641], classification [707, 739], autoregression [758], and Bayesian statistics [162].

Tab. 2.3: Data management tasks that can be optimized by machine learning algorithms.

Task	Explanation
Data cataloging and curation	To override manual data labeling by using automatic labeling [202]
Data preprocessing, anomaly detection	To identify missing data, help identify and fix incorrect labels [620, 641, 758]; to identify observations or data elements that raise suspicions because they significantly deviate from the majority of the data [641]
Data mapping	To match fields from multiple datasets into a manageable and harmonized system [66]
Feature engineering	To create candidate features out of a dataset from which the best can be selected and used for training [221]

2.1.6 Challenges for Machine Learning Methods in the Context of Health Care

Computational innovations are bound to transform health care. Nevertheless, there are a number of challenges that have to be tackled in order to successfully adopt machine learning in the clinical setting. The challenges concern data quality (e.g., missing data, outliers), learning while preserving privacy, the interpretability and generalizability of machine learning models, and clinical implementation. In the following section, we

will elaborate on some of these challenges and potential mitigation strategies.

Missing Data One of the most common machine learning challenges faced is the occurrence of missing data in digital health datasets. There is a multitude of reasons why missing data occur, ranging from (human) data-entry error, missing measurements, dropouts in clinical studies, and merging unrelated data, to software errors in the data processing pipeline [285, 646, 663]. Missing data can have a significant effect on the data quality, lead to application performance degradation, cause analytical issues, and bias outcomes. In the context of medicine, the latter has been previously associated with misdiagnosis, wrong treatment decisions, and even discrimination of marginalized groups [656, 744]. Moreover, most state-of-the-art machine learning models require complete input variables. Missing data is typically handled by either the deletion of all data for an observation that has one or more missing values or the replacement with estimated values (i.e., imputation) [177]. A variety of methods have been developed that can account for different levels of sparsity in the data as well as efficiently handle missing information (Figure 2.1). Popular machine learning algorithms include k-nearest neighbors [50], multi-task Gaussian processes [729], random forest–based approaches [647, 670], matrix factorization [341, 692], discriminative deep learning methods [64], and generative deep learning methods [469, 596, 749]. Another elegant strategy of handling missing data is the use of end-to-end models that impute and predict jointly, such as Gaussian process adapter [393] and interpolation-prediction networks [610], and models that do not require imputation and can act on irregular data directly, including attention models and gated recurrent unit-decay [702, 725]. For a comprehensive review on the problem of missing values and strategies for handling missing data, see [188].

Outlier Detection Another noteworthy challenge is how to detect and handle outliers in a dataset. Outliers are defined as observations that raise suspicion because they deviate markedly from other observations in the given dataset. Common causes of the occurrence of outliers in digital health datasets include, measurement error, data entry error, sampling error, and natural outliers. A special category of outliers are the intentional outliers, which are dummy outliers created to assess the efficiency of detection methods. It is important to note that outliers are inherently different from noise, which is commonly defined as a random error or variance. The outlier is part of the data and can even carry (clinically) important information, while noise is simply a random error (e.g., mislabeled data, missing data). Detecting outliers in a dataset is a highly relevant task as outliers can impact the distribution of the data, increase the error variance, reduce the power of statistical tests, introduce bias, influence estimates, and impact key assumptions of statistical tests. A multitude of statistical and

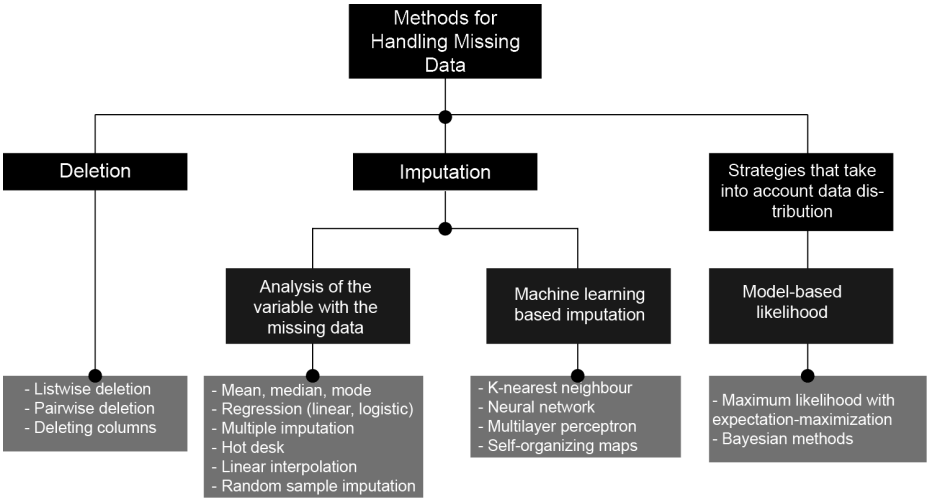


Fig. 2.1: Strategies for handling missing data.

machine learning methods exists for the task of outlier detection, including k-nearest neighbor [290], linear regression, naive Bayes [541], decision trees [620], and support vector machines [764]. The classic distance-based methods are empirically highly successful [104, 330, 759]. For instance, they might deem certain patients outliers if they are distant from other patients in the dataset. A patient is deemed an outlier if they are distant from a randomly drawn subset of historic patients (Figure 2.2). This scheme is extremely scalable, as it requires only the computation of distances between the patient and the small subset of historic patients, even for large clinical data warehouses. The size of the sample can even be explicitly optimized.

Learning while Preserving Privacy Data privacy has become one of the most important issues of our time. A breach of personal information can infringe fundamental rights and freedoms of an individual, including the risk of being identified and disclosure of personal (health) data. Data privacy breaches, such as the Facebook Cambridge Analytica scandal [287], have made patients and their caregivers reluctant to share sensitive and personal information. In response, data-privacy concerns have taken center stage and countries around the world have implemented legislation, such as the European Union's General Data Protection Regulation (GDPR) [706] and the California Consumer Privacy Act (CCPA) [488]. Medical questions that are tackled by a data-driven approach often require access to sensitive, personal information.

Major efforts have been undertaken to develop privacy-preserving machine learning algorithms that keep the patients details secure without compromising the model's performance [533, 719, 736]. Federated learning [394, 745] addresses some of these

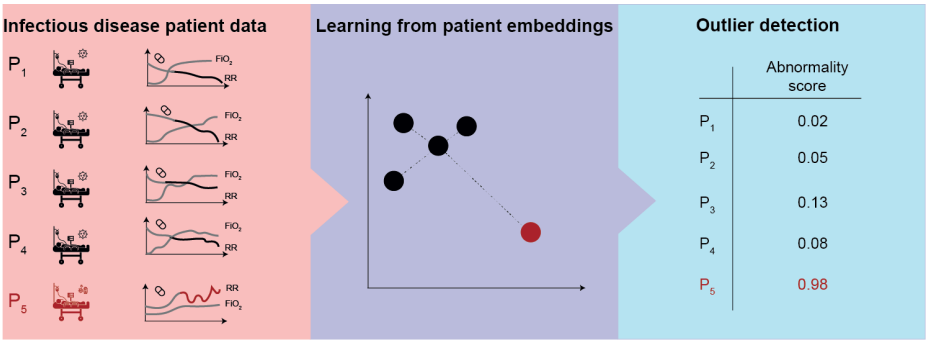


Fig. 2.2: Illustration of the goal of abnormality detection in COVID-19 patients (left panel). The electronic health record of a new patient P5 is checked for being abnormal relative to all patients in the clinical data warehouse (here patients P1-P4) (center panel). Relevant features are extracted to compare the similarity of the new patient to previous patients. The features describing a patient are referred to as patient embeddings (right panel). Based on the (dis)similarity of the new patient to previous patients, an abnormality score is computed. In this example, the abnormality score is the distance to the most similar patient in the database, which is largest for P5 (0.97).

concerns through training algorithms collaboratively without requiring exchange of raw data. In federated learning, the model parameters are handled centrally. To overcome this "concentration of power", swarm learning was recently introduced [719]. The principle of swarm learning is to build the machine learning models independently on data from individual sites (e.g., hospitals) and share the model parameters via a so-called swarm network. With this approach, swarm learning secures data sovereignty while preserving privacy and confidentiality. Data mining is another popular discipline in medical data science concerned with preservation of privacy [12, 18]. Also known as knowledge discovery in data, data mining is the process of automatically uncovering novel patterns and trends in big data that would otherwise remain hidden [137]. In the recent years, data mining has been successfully applied in a variety of medical disciplines: detection of diabetes [36], cancer prognosis prediction [572], biomarker discovery [244], sepsis [91], and prediction of stroke mortality [186]. As data privacy should be preserved at all costs, numerous privacy-preserving data-mining methods have been developed. These include randomization [183], classification [130], clustering [229], association rule [539], K-anonymity [134], L-diverse [425], distributed privacy preservation [182], condensation [10], and cryptographic [376, 501]. A comprehensive survey on the contributions of privacy preserving data mining techniques can be found in [16, 567].

Access to data generated within the healthcare systems is often restricted due to privacy and confidentially concerns. A strategy to make sensitive health data available is to de-identify or anonymize the data by deleting or encoding identifiers that link individuals (e.g., names and patient identifier), by perturbating the data (e.g., applying round-numbering methods and adding random noise), by swapping data (e.g., shuf-

fling dataset attribute values), or by generalizing information (e.g., grouping variables). Even with these de-identifcation efforts, it remains extremely difficult to guarantee that the re-identification of individual patients is not possible. A promising mitigation strategy is to generate synthetic, representative, data that can be safely shared. Synthetic data can both be used to augment datasets and generate artificial, but realistic patient data that can be shared, even across geographical and political borders. For this form of synthetic data generation, Generative Adversarial Networks (GANs) [235] are well suited. Briefly, GANs are a type of deep learning model that consists of two networks one called the generator and the other called the discriminator. These two networks are simultaneously trained competitively, as in a zero-sum game framework. The generator learns how to map from a latent space to a data distribution of interest, i.e., generate candidates of synthetic patients, and the discriminator evaluates the candidates distinguishing from the true distribution. In the context of clinical data, medical GAN (medGAN) [117] is a recent approach that can generate high-dimensional discrete variables via a combination of variational autoencoders (VAEs) and GAN. Furthermore, medical Wasserstein (medWGAN) and boundary-seeking GAN (medBGAN) improved the performance of medGAN to generate synthetic data from the "Medical Information Mart for Intensive Care" database and the Taiwan National Health Insurance Research Database [144]. Instead of a general GAN, medWGAN uses an improved generative network named WGAN-GP, where the model overcomes the issue of fails to converge in some settings owing to the use of the weight-clipping technique using gradient penalty [43]. Finally, medBGAN improves GAN training to create new samples that lie on the decision boundary of the discriminator at each update [43].

Interpretability and Generalizability of Models In addition to being applied in complex high-stakes settings such as medicine, machine learning algorithms are also becoming increasingly complex in terms of their architecture. At times, these algorithms become so complex that the humans forfeit comprehension of the underlying models or how variables are jointly related to make predictions. Modern machine learning algorithms are commonly referred to as "black boxes". The alleged black box nature constitutes a major barrier to the adoption of machine learning in the clinical routine. But what is the reason for not trusting a machine learning model that has been proven to perform well and can accurately diagnose patients? "The problem is that a single metric, such as classification accuracy, is an incomplete description of most real-world tasks" [179]. The interpretability of machine learning models is critical to understand the accuracy of findings, identify variables that drive the predictions, improve model performance, guarding against embedded bias, and debug models. Medicine is among the domains where scientists are often compelled to implement simpler and interpretable machine learning models (e.g., linear models or decision trees) as every decision being taken by the model has to be interpretable. Clinical staff needs to understand why a

machine learning algorithm generates the results it does (i.e., interpretability) and ideally how it arrives at its conclusions (i.e., explainability) [190]. Better interpretability might come at the expense of model performance. As biological associations are hardly ever of a linear nature, complex models, including ensembles and neural networks, typically result in more accurate performance. Incomplete interpretability is often compensated with judgement, knowledge provided by domain experts (e.g., clinicians), rigorous monitoring, and diligent understanding of the data used. The development of methods to enhance the interpretability of machine learning models is a vivid area of investigations. As a matter of fact, numerous model-agnostic interpretation tools exist that can be applied to any supervised machine learning model [529]. There are two major categories of model-agnostic methods. First, local methods that describe individual predictions and secondly, global methods that explain how features affect the prediction as a whole. Table 2.4 provides an overview of common local and global model-agnostic methods. These model-agnostic interpretability methods allow researchers to interpret the results of (complex) machine learning models and can pave the way for the implementation of such models in the clinical routine.

Tab. 2.4: Common local and global model-agnostic interpretation methods of machine learning models. For a comprehensive overview on interpretable machine learning, see [453].

Local methods	Global methods
Individual Conditional Expectation (ICE) [233]	Partial Dependence Plot (PDP) [213]
Local Surrogate (LIME) [527]	Accumulated Local Effects (ALE) Plot [25]
Counterfactual Explanations [711]	Feature Interaction [214]
Scoped Rules (Anchors) [528]	Functional Decompositon [281]
Shapley Values [661]	Permutation Feature Importance [102]
SHAP (SHapley Additive exPlanations) [661]	Global Surrogate [152]

Apart from being interpretable, generalizability is a desired attribute of machine learning algorithms. Generalizable refers to the ability of a trained algorithm to perform well on unseen data. One particularly elusive challenge regards generalizability across different patient groups [489]. There are numerous examples of promising machine learning applications that struggled when applied to diverse populations. Google, for instance, introduced a machine learning algorithm for the diagnosis of diabetic retinopathy that performed poorly in India [7]. This is likely attributable to the fact that the algorithm was developed, trained, and evaluated on a dataset that lacked the necessary ethnoracial and demographic diversity. Another example of unintended effects of artificial intelligence is an algorithm that was developed to detect skin cancer on images of skins. While the algorithm performed well on fair skin, it was not able to reliably diagnose lesions on darker skin [240].

These examples highlight the importance of data from diverse groups (i.e., in terms of sex, ethnic background, and race) are fundamental to realize universal precision medicine. The reality, however, is that data is often derived from a worryingly small and homogeneous sample of the population (e.g., white male individuals). As a result, ethnoracial disparities are evident in many patient populations and include differences in access to care, time to diagnosis, treatment, and mortality [545, 566, 748]. In order to eliminate bias and create fairness and equity, scientists have to be conscious of bias that can occur at different levels, namely data collection and selection, model development and evaluation, as well as model deployment and clinical implementation. Data is driving force behind any machine learning and artificial intelligence algorithm. That is why, the data underlying the development and evaluation of algorithms must be unbiased and representative of the target populations to avoid generating or perpetuating biases that may worsen patient outcomes. Often bias is rooted in systematically skewed data collection, e.g., through clinical trials predominantly carried out with white male participants, or the reliance on historical data that might have been subject to biased data generation or clinical practices. To mitigate bias in the data, diverse and well balanced study populations are crucial for any collection and/or selection of data (e.g., clinical trials, registry, electronic health records). Particular attention should be paid to ethnoracial diversity, sex/gender balance, socioeconomic equity, and other social, and ethical, determinants of disease and access to healthcare. Assuming that the available data is unbiased, researchers have to carefully select the data variables to avoid introducing a bias in the phase of algorithm development.

If possible, algorithms can be tested on different patient populations for both scientific and ethical performance. Ideally, the development, evaluation, and clinical implementation of algorithms is done in liaison with clinicians to ensure that the algorithms do not exhibit bias in the clinical setting. Lastly, the healthcare system, in which machine learning tools are implemented, is an important entry point of bias. Awareness of inherent biases of machine learning assisted tools among healthcare professionals is pivotal to mitigate bias. This starts by ensuring equitable patient access to the technology, and then diligently observing how it performs in diverse populations and underrepresented communities. Any bias noticed should immediately be reported to an appropriate committee at the hospital, which can then communicate with the developers. Understanding bias inherent in medical technology allows clinicians to question the accuracy of the technology if the results do not meet the expectations from their clinical expertise.

Clinical Implementation and Validation Considerable technological progress has been made over the last decades with machine learning applications transforming the clinical decision making and how health resources (e.g., data) are managed. Nevertheless, the greatest challenge of machine learning applications is not the medical utility,

but rather the implementation in the clinical routine. While these developments are crucial to advance health care, they raise a number of ethical and legal ethical concerns that machine learning and artificial intelligence might harm patients and/or clinicians. Technological and diagnostic failures can lead to adverse events or seriously harm patients. It is not yet clear who is liable for malfunctions in, or erroneous decisions made by artificial intelligence-based clinical tools that result in inaccurate or delayed diagnosis [510]. The attribution of accountability becomes even more complicated when an artificial intelligence-based clinical tool gives a wrong treatment recommendation, yet the clinician makes the final decision. Is the clinician liable or can the liability be delegated to a company or person that engineered the tool? In [510], Price et al. provide an overview of potential scenarios and associated probable legal outcomes related to AI use in clinical practice. Under the current law, clinicians are shielded from liability as long as they do not deviate from the standard of care [532]. As a consequence, clinicians are advised to utilize machine learning-guided tools to support and confirm existing decision-making processes as opposed to solely relying on computational algorithm output for diagnosis or treatment selection [441]. The complexity further increases when considering that there are multiple stakeholders in the ecosystem of liability, including the healthcare institutions that purchase and implement computational algorithms.

2.1.7 Future Directions of Machine Learning in Medicine

The accelerating generation of unparalleled amounts of health data will lead to fundamental changes in medicine and health care. Machine learning applications are poised to play an increasingly prominent role in medicine. Specifically, they will facilitate early disease recognition, refine diagnosis and prognosis, support therapy decisions, and streamline biomedical data management. Importantly, machine learning-guided systems will not replace clinicians or therapists, but will augment their efforts and time to care for patients thanks to guidance for clinical decision-making and the automation of time-consuming and repetitive tasks. As of yet, many barriers exist to the adoption of such applications in the clinical routine. In the coming years, the data explosion will continue and reach unparalleled dimensions, including the number of patients (e.g., UK Biobank), the length of time series (e.g., ICU monitoring data, wearable devices), and the breadth of data type (from molecular to higher-level phenotypes). This affluence of rich datasets will open new avenues for machine learning-driven applications to assist in clinical decision-making. In addition to refining clinical processes, machine learning and artificial intelligence will also play a pivotal role in other important areas of biomedical research, such as protein structure prediction, molecule design, drug discovery, or single-cell research. A groundbreaking example is the recently introduced machine learning approach *AlphaFold* [308], which performs predictions of protein structure with unprecedented accuracy, by incorporating physical and biological knowl-

edge. Besides the technological advances and the availability of vast amounts of data, overcoming the challenges of handling missing data and outliers, preserving privacy, interpretability, and clinical application will be critical for the adoption of machine learning-supported guidance tools in clinical routine.

2.2 Virus Detection

Roland Hergenröder
Frank Weichert
Konstantin Wüstefeld
Victoria Shpacovitch

Abstract: Amid the accelerating spread of viral diseases throughout the world, the rapid detection of pathogens is of essential importance. Viruses can be transmitted very quickly via contacts in public transportation or at large social events. Therefore, a rapid virus test system for such crowded locations is highly desirable. The plasmon-assisted microscopy of nano-objects (PAMONO) sensor is one such analytical instrument. The sensor required the development of software and optomechanical parts to detect viruses in complex biological liquids. While the focus lies on viral particles, other nanoparticles can also be analyzed by employing a similar principle. The latter issue vastly expands the potential application field of the PAMONO sensor. The developed methods are tailored to the spatiotemporal characteristics of the underlying sensor system, making use of the adaptivity of machine learning approaches. As a result, 80 nm to 300 nm particles can be detected in signals with different types of imaging artifacts and different resolutions, reaching accuracies of over 80 % with respect to the expected particle counts of test samples. For mobile use as a rapid test system, resource-saving and real-time capability are of similar importance to make the device accessible in as many application areas as possible. Multi-objective optimization in terms of detection quality and energy consumption was applied to demonstrate that the usage as a mobile system is feasible.

2.2.1 Introduction

For the detection of nanoparticles such as virus particles a device that can make them perceptible is required. Such a device is the PAMONO sensor [385, 607], which pumps a liquid or air sample through a flow cell to reveal the particles of interest contained in it. This is achieved by making use of what is known as the Surface Plasmon Resonance (SPR) phenomenon [340].

Special proteins—antibodies—immobilized on a gold sensor film help to accomplish the specificity of viral particle detection. Particles that bind to the antibodies cause local changes in the reflection conditions near the surface of the film. Due to locally increased reflection, particle binding events become detectable by a charge-coupled device (CCD) [389] integrated into the PAMONO sensor.

The limits of a manual analysis are reached quickly, when trying to identify particle signals. Even after enhancing the visibility of particles in the recorded images by preprocessing, finding particle regions is a tedious and time-consuming task for a human observer. Test analyses determined that it takes an expert approximately two days to accurately analyze a dataset for an experiment of around 4000 images with varying results for different particle sizes, different levels of disturbances, and different human observers [613]. This time range, the need for visually trained experts, and the deviations in the subjective perception of different persons predestine this task to become the subject of automated analysis to enable the PAMONO sensor to be used as a rapid test.

To reduce manual interactions and to enable quick testing by non-expert users, more adaptive solutions from the field of deep learning were developed to adapt to specific signal characteristics while tolerating deviations that inevitably occur between different recordings when operating outside a controlled environment.

While manual interactions are shifted from on-site usage to training time, these approaches make actual on-site testing faster. The challenge which arises in return is the high amount of manually annotated training data to learn the patterns of interest. At the same time, the recording and annotation of new experiments cause material and time costs. This is a problem that is worth addressing since it can be observed in various tasks of medical data analysis. Dealing with the limited availability of training data while leveraging the generalization of machine learning is, therefore, a key aspect in this area.

There are three major challenges to be addressed when detecting nanoparticles in samples: dealing with varying artifact characteristics and intensities, real-time capability, and the feasibility of mobile usage. Considering the on-site operation, the concept of mobility again contains the aspect of resource optimization in terms of computing power and energy consumption. We present an approach for multi-objective optimization of parameters in the employed algorithms. This optimization can target high detection accuracy or low energy consumption. Alternative approaches based on deep learning overcome the need for defining specific operators by learning them from more general functions. The aspects of the application under natural conditions, including the analysis of physical particle sizes, are viewed with particular attention. The analysis of physical particle sizes is also described in that context. It can provide a more accurate classification of the contained particles and enables plausibility checks by taking domain knowledge about the specific types of particles into account.

Here is a short overview of the sections below: Section 2.2.2 provides details on the different types of particles. The setup of the PAMONO sensor is detailed in Section 2.2.3. Section 2.2.4 describes the underlying data characteristics and the classic and deep learning-based methods for the detection of nanoparticles. Section 2.2.5 presents an approach developed for the multi-objective optimization of parameters used in an operator. Section 2.2.6 describes aspects of the application in natural environments with particular emphasis on determining the size of the analyzed particles and an ap-

proach that increases the robustness of image analyses based on generative adversarial networks. Finally, Section 2.2.7 provides an outlook on potential approaches to improve the hardware and software of the PAMONO sensor.

2.2.2 Types of Detectable Nanoparticles

It is first worth noting which types of particles the PAMONO sensor can analyze. Different physical and biological characteristics can be spotted, that are attributable to biological Nanoparticles (bio-NPs). Besides particles of interest such as viruses, Virus-Like Particles (VLPs) [754], and Extracellular Vesicles (EVs) [504, 505], there are interfering objects can also be observed, such as lipid and protein agglomerates, which are often considered contaminating substances hampering the bio-analytical examination of samples.

While viruses are well-recognized as the smallest infectious agents containing only one type of nucleic acid, Ribonucleic Acid (RNA), or Deoxyribonucleic Acid (DNA) [332], VLPs, and extracellular vesicles are significantly less analyzed. It is important to highlight the key difference between VLPs and viruses: VLPs do not possess any nucleic acids and, thus, lack a principal opportunity to reproduce themselves in the host organism. On the other hand, VLPs carry the same antigens (molecules considered foreign by the immune system) on their surface as the corresponding native viruses [754]. Thus, VLPs in science can serve as a safe and reliable model of dangerous viruses since VLPs efficiently mimic the structural properties of corresponding viruses but cannot replicate. In practice, VLPs are well known as commercial medical products serving as a basis for vaccines [754].

Another group of bio-NPs, Extracellular Vesicles (EVs), have recently started to attract the attention of scientists and physicians. EVs are submicro- and nano-sized vesicles released by the majority of cells [505]. Another principal feature of EVs is their ability to carry different molecules inside as well as on their surface. Among such active molecules are hormones, growth factors, active peptides, and nucleic acids [505]. Their cargo makes EVs active messengers participating in intercellular communication and reflecting cellular status under normal conditions or during the pathological processes. Moreover, the abundance of EVs in body fluids such as blood or saliva drew the attention of clinicians and medical researchers, who harness EVs as a means of drug delivery or to estimate their potential as biomarkers of the progression of a disease.

However, any analysis of bio-NP samples for scientific and practical needs requires the selection of reliable techniques and instruments. Certainly, bio-NPs have to be swiftly characterized for their abundance in a sample and their size. From a different perspective, biochemical information regarding their surface antigens (proteins) and their content is of interest as well. A simultaneous quantification and determination of the sizes of bio-NPs can be achieved by the principle of surface plasmon resonance (SPR). The plasmon-assisted microscopy of nano-objects (PAMONO) sensor, which exploits

this principle, is an instrument for label-free and specific detection of individual bio-NPs in solutions [245, 772]. In the following section, the sensor and the underlying principle for visualizing particle signals are introduced in more detail.

2.2.3 PAMONO Sensor

Surface plasmons can be thought of as propagating electron density waves. These waves can be excited by incident light (usually by an incident laser beam) in a thin metal film at a dielectric-metal interface. It is precisely this thin metal film that serves as a sensor surface. Surface Plasmon Resonance (SPR) is a physical phenomenon that served as a basis for the development of the PAMONO sensor.

Conventional SPR biosensors deal with measurements of the layers of bio-molecules formed onto the sensor surface, and thus, conventional SPR sensors are not applicable for the detection of individual Nanoparticles (NPs). By contrast, the special quality of the PAMONO sensor is exactly the ability to detect the binding of individual NPs to the gold sensor surface [772]. Kretschmann's scheme of plasmon excitation is utilized in the PAMONO sensor as well as in the majority of conventional commercial SPR-based biosensors [350]. However, there are specific issues that distinguish the PAMONO sensor from known conventional SPR biosensors. In Kretschmann's scheme, shown in Figure 2.3, a p-polarized light (polarization of the electric field occurs in the plane of incidence) illuminates a glass prism with a very thin (tens of nanometers) noble metal film deposited on the base of the prism. Often a superluminescent diode or a diode laser is used as a source of light [350]. Surface Plasmons (SPs) are excited as propagating electron density waves at the metal-dielectric interface in the presence of p-polarized incidence light at a particular angle [328, 577, 693]. This event occurs when the energy of an incidence beam transforms into electron-polaritons within the thin metal film deposited on the prism. SPs excited along the metal-dielectric interface result in a substantial reduction of the reflection intensity. In turn, this fact leads to changed reflection conditions, which are extremely sensitive to any refractive index changes occurring close to the metal-dielectric interface [577]. Such changes can be caused by the adsorption of molecules onto the metal film surface. SPR-based biosensors harness this trait and enable the analysis of interactions between bio-molecules immobilized onto the metal film surface and their counterparts in the analyzed liquid sample. Such analysis can be performed in real time and without labeling the target molecules. Thus, it is not surprising that conventional SPR biosensors are actively used to measure binding constants and the kinetics of bio-molecular interactions, and to perform concentration measurements [577].

The PAMONO sensor also harnesses the most convenient scheme of plasmon excitation: Kretschmann's configuration [350]. Figure 2.4 shows a photograph of the device setup and the flow cell as the core of the apparatus. The entire device fits into a suitcase-sized enclosure.

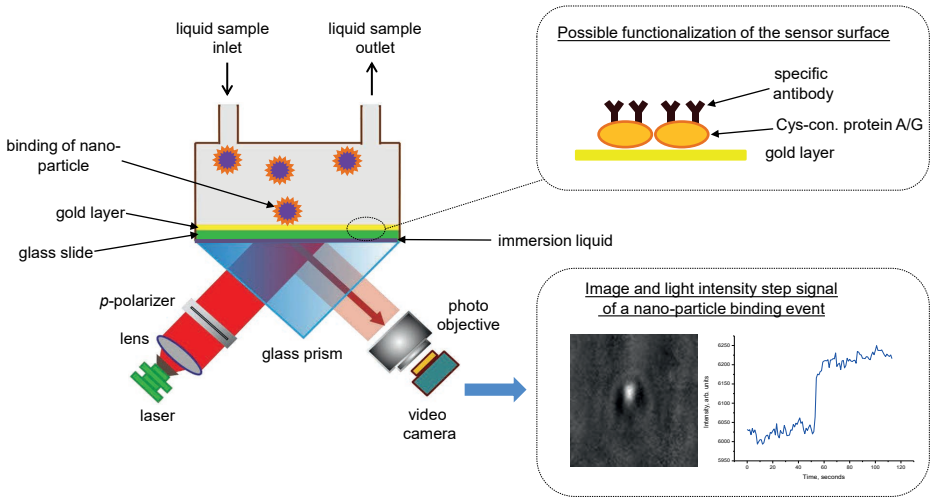
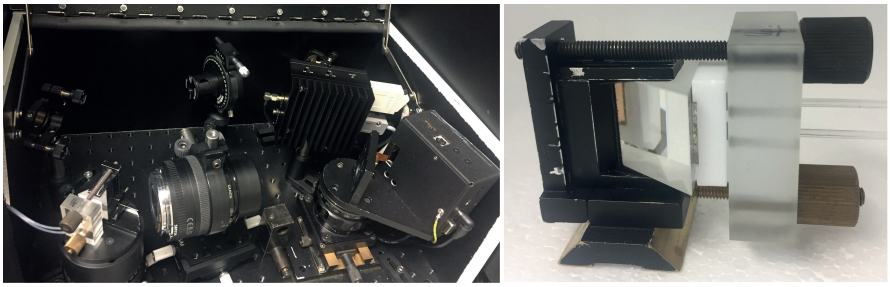


Fig. 2.3: Schematic setup of the PAMONO sensor (left), abstract view of a used antibody coating (top right), preprocessed image of an attached particle clearly visible as a bright elliptic region (bottom right, left image), and the average pixel intensities of this area over time (bottom right, right image); modified from [669].

Moreover, the events preceding the substantial reduction of the reflected light as well as the events leading to the restoration of reflection are similar for the PAMONO sensor and conventional SPR-based sensors. In the case of the PAMONO sensor, such events occur locally, in the spot of NP binding, not on the entire sensor surface as it happens in the case of classic SPR sensors [773]. A developed model [773] explains key differences in physics between the detection of bio-molecule layer formation (classic SPR sensor) and individual NPs (PAMONO sensor). Polystyrene NPs were employed as a model system [245]. The use of these particles helped to demonstrate linear dependency between the number of signals detected by the PAMONO sensor and the concentration of particles in liquid samples. In turn, this fact confirmed the applicability of the PAMONO sensor for the concentration measurements of NPs, in which NP concentration is expressed as a number of particles in a volume unit [245]. The work of Shpacovitch and colleagues [608] focused on the bio-analytical features of the PAMONO sensor and demonstrated the ability of the PAMONO sensor to detect not only HIV-VLPs (100 —-140 nm) but also influenza A viral particles (80 —-120 nm).

The selectivity studies were performed in phosphate buffered saline (PBS buffer) employing specially engineered HIV-VLPs of two types: one containing target protein on the surface and one lacking it [608]. Under these conditions, the selectivity of HIV-VLPs binding to the PAMONO gold sensor surface reached 90 % without special treatment of the sensor surface with substances preventing the binding of non-target VLPs [608]. Moreover, the ability of the PAMONO sensor to work with biological samples containing serum was also demonstrated [608].



(a) The real setup of the PAMONO sensor corresponding to the schematic setup shown in Figure 2.3. The case of the instrument is approximately the size of a suitcase.

(b) The flow cell with mounted gold-coated glass plate attached to a prism base. The tubing system serves to guide a liquid sample into and out of the flow cell.

Fig. 2.4: Photos of the PAMONO sensor (a) as a whole and of the flow cell (b) as the heart of the device individually.

Further work [609] proved the power of the PAMONO sensor for the detection of the other type of bio-NPs: extracellular vesicles. The authors employed cysteine-conjugated protein A/G for the functionalization of the PAMONO sensor surface. This was done to allow for the elution of bio-NPs captured on the sensor surface and, thus, enable a post-PAMONO analysis [609]. Moreover, the PAMONO sensor was capable of supplying sufficient information for the sizing of studied polystyrene nanoparticles. Such information could be extracted from the intensity step signal caused by NP binding. It is important to mention that the Nanoparticle Tracking Analysis (NTA) instrument Malvern Panalytical NanoSight LM10[1] was used as a reference method in the studies performed with the PAMONO sensor. Thus, it was also necessary to verify the accuracy of the LM10 instrument before its use in the studies. This work was performed by Usfoor and colleagues [696]. It was demonstrated that NP size measurements performed by the LM10 device are quite accurate, but concentrations were not determined precisely. Moreover, the NTA analysis of bio-NPs requires the labeling procedure of target particles, while the PAMONO sensor provides results employing a label-free approach. In detail, the drawbacks and advantages of the PAMONO sensor and other SPR-based platforms for the sizing, quantification, and biochemical analysis of extracellular vesicles are given in a review work [606]. One of the advantages of the PAMONO sensor is the possibility of NP quantification without prior calibration, as as shown by Kuzmichev and colleagues [361].

The analysis of sensor data requires the use of robust detection algorithms that can adapt to data variations in real use cases. Approaches that incorporate this criterion are presented in the following sections.

1 Malvern Panalytical NanoSight LM10, https://www.malvernpanalytical.com/en/products/product-range/nanosight-range/nanosight-lm10, accessed 31 March 2022.

2.2.4 Automated Nanoparticle Detection

Based on the challenges of nanoparticle detection, a signal model was developed to represent the individual components of the total signal

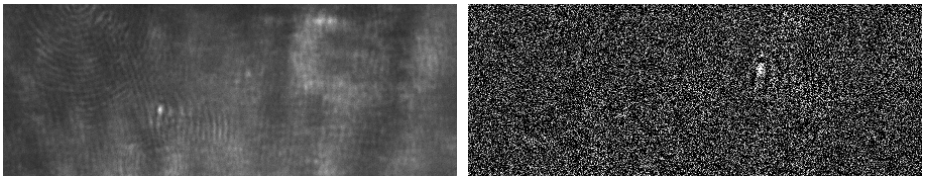
$$I(x, y, t) = B(x, y) \cdot A(x, y, t) \cdot P(x, y, t) + R(x, y, t) \tag{2.1}$$

which consists of the background signal B, which is constant within a recorded image, an artifact signal A, the signal of interest P, and a residuum R, which includes random noise [614]. Figure 2.5 shows an example of an unprocessed recording and the corresponding image in which the contained particle signal is made visible in a *preprocessing* step by removing the background B and reducing the noise R. This is achieved using a sliding-window approach that averages signal values, amplifies pixel intensities increasing over time, and weakens constant or falling intensities. Subsequently, a dynamic contrast enhancement is applied in some approaches to further emphasize particle signals [733]. After that, the regions of interest can be spotted as elliptical areas that are brighter than their environments. Based on the presented signal model, the goal of the automated detection is to make the pixel values of signals of interest P more distinguishable from their surroundings. Typically this is done by employing approaches that attempt to highlight particles directly and, in some cases, by weakening artifact signals beforehand.

From the algorithmic point of view, the detection of nanoparticle signals on preprocessed images can be seen as a combined task consisting of blob detection [639] and time series analysis as particles can only be reliably recognized when their spatial features like region size and shape as well as their temporal behavior are taken into account. The characteristic temporal behavior, which can be described as a step-like curve, is shown in Figure 2.6 for two images from sets with different particle sizes. From the algorithmic point of view, the detection of nanoparticle signals on preprocessed images can be seen as a combined task consisting of blob detection [639] and time series analysis. This is because particles can only be reliably recognized when their spatial features like region size and shape as well as their temporal behavior are taken into account. The characteristic temporal behavior, which can be described as a step-like curve, is shown in Figure 2.6 for two images from sets with different particle sizes. The origin of the temporal characteristics lies in the way a particle of interest interacts with the antibody layer. When such a particle attaches to it, it remains in contact for a prolonged time. It is assumed that, in the time of one recording, particles of interest stay attached permanently while other particles only cause short-time peaks in the measured intensities as their physical shape does not match the specific antibodies applied to the gold film (see Section 2.2.3).

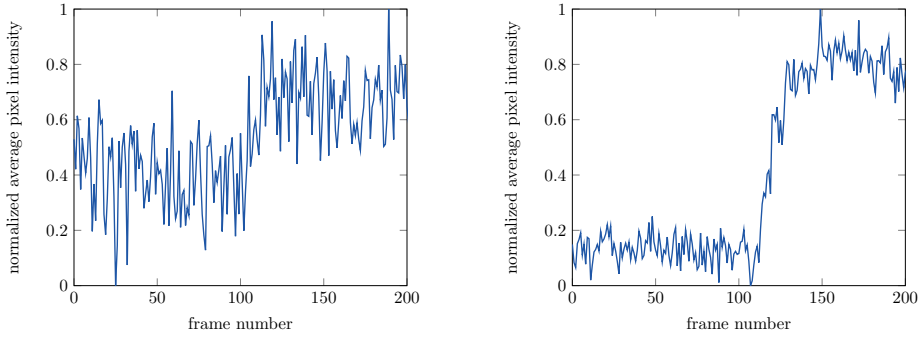
A closer look at intensity curves belonging to particles of different sizes reveals that a larger particle causes a higher intensity difference

$$I(x, y, t_j) - I(x, y, t_i), t_i < t_j$$



(a) Unprocessed **(b)** Preprocessed

Fig. 2.5: Comparison of (a) an unprocessed, recorded image and (b) the same image after preprocessing, which enables detecting particles as ellipsoids with high pixel intensities compared with the background. The particle that can be seen in the preprocessed image can hardly be detected in the unprocessed image data. To improve the visibility of particles, the preprocessing has to make use of temporal information from a time window around the current frame.



(a) Intensity of an 80 nm particle over time. **(b)** Intensity of a 200 nm particle over time.

Fig. 2.6: Exemplary comparison of two signals over time belonging to different particle sizes.

for a position (x, y) and times t_i, t_j than a smaller particle. With smaller particle sizes, the particle signal P becomes weaker, while inaccurate adjustments of the optical instruments, less clean samples, and increasing external influence imposed on the device increase the intensities of the artifact signal A and the residuum R. This results in a lower signal-to-noise ratio (SNR) [127]

$$SNR(P, Q) = \frac{|\mu(P) - \mu(Q)|}{\sigma(P)} \tag{2.2}$$

that is determined based on the particle signals P and the non-particle signals $Q := A \cup R$, where μ is the average value and σ is the standard deviation of a set of intensity values. While not suitable for analyzing structured artifacts, the measurement can illustrate the visibility of particles in an image that predominantly contains random noise besides the particle signals. A low SNR value indicates a bad detectability of particles caused by a low intensity of particle signals P or a high intensity of random noise R. Specifying a metric indicating the strength of the influence of structured artifacts would require

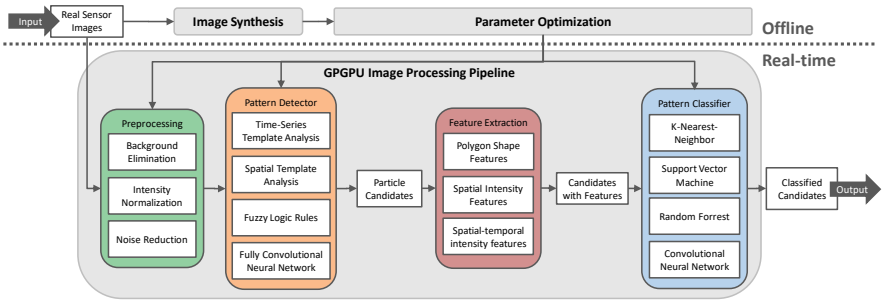


Fig. 2.7: An abstract processing pipeline for PAMONO images showing different methods that can be used for preprocessing, pattern detection, candidate feature extraction, and classification [385].

a more complex definition that is less interpretable. For this reason, the SNR value is used when a directly understandable indication of particle visibility is desired.

2.2.4.1 Stages of Detection

With respect to the characteristics described in Section 2.2.4, different methods exploiting spatial and temporal features for nanoparticle detection were evaluated. Figure 2.7 shows the stages of an abstract detection pipeline containing alternative approaches for the different stages. In general, the task of automated nanoparticle detection can be divided into different subtasks. The first step of any presented detection approach is preprocessing with the goal of image restoration based on a signal model. In addition, image enhancement techniques are used to improve the distinguishability between particles and other signals. With the preprocessed images at hand, a segmentation of the image areas and the detection of candidate regions takes place, which is summarized as the *pattern detector*. Then a *feature extraction* method determines features that are used in a *pattern classifier* to check the properties of candidates so that non-particle candidates can be filtered out.

The whole pipeline can be optimized for a specific criterion using the offline parameter optimization described in Section 2.2.5.1.

For preprocessing images, a sliding-window method with a fixed window size in different variations [399, 733], as well as a constant background removal method [613], have shown to be effective in different approaches. Depending on the downstream algorithm for detection, additional noise reduction techniques such as Gaussian or median filtering [613], wavelet denoising [401], brightness correction [399], or dynamic contrast enhancement [733] are used to provide a better separability between particles and other signals.

While a variety of approaches is applicable for subsequent tasks, different approaches offer diverse strengths and weaknesses for different data characteristics. The

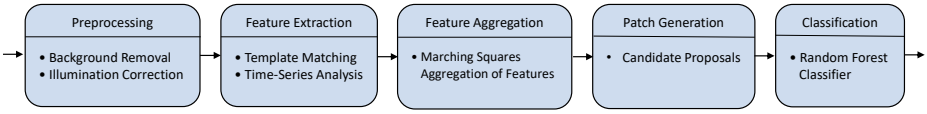


Fig. 2.8: Example of a classic pipeline for the detection of nanoparticles based on template matching and time-series analysis for the generation of particle candidates and the evaluation of polygon shape features in a random forest approach for classification (adapted from [399]).

following are available operations for the subtasks of particle detection. An appropriate selection of operations is presented in Section 2.2.5.1.

For the task of pattern detection, (fuzzy) template matching in different variants [399, 400, 401, 613] and convolutional network approaches [387, 733, 746] were developed. With particle candidate regions generated by the pattern detector, feature extractors obtain spatial or spatiotemporal features, which are then evaluated by a pattern classifier. Extractors generate, for example, polygon shape features like the covered area or the circularity of a polygon [399] or measures describing representations in other spaces like the Fourier or wavelet space [746].

Evaluated classifiers are, for example, k-nearest-neighbor [615], support vector machines (SVM) [615], random forests [387], and convolutional neural networks [387, 733].

In the end, a connection between regions to single particle traces distributed over consecutive frames is established. This is particularly important when counting the detected particles is desired instead of just detecting if particles are present at all.

A concrete example of a classic detection pipeline is shown in Figure 2.8. It utilizes template detection and fuzzy time series analysis [399] for pattern recognition and classifies particle candidate patches with random forests based on polygon shape features.

Template matching, which has proven effective in various examples, uses a previously recorded particle region as the predefined template patch T to detect similar regions of the same size in the current image I. This is done by calculating a normalized cross-correlation

$$R(x,y) = \frac{\sum_{x',y'} T(x',y') I(x+x',y+y')}{\sqrt{\sum_{x',y'} T(x',y')^2 + \sum_{x'} \sum_{y'} I(x+x',y+y')^2}}.$$

for each pixel position (x,y) in the image with the template patch [100, 399]. In simplified terms, the template patch is moved over the image while the correlation of the template patch with the underlying image area is determined at each position.

Although using classic methods like template matching with optimally adjusted parameters can lead to high detection accuracies, slightly outperforming some neural network approaches [385], it has to be noted that this is only possible if the parameters are individually optimized for each change in the setup and, in the worst case, for each dataset [385, 613]. This is a disadvantage for use as a rapid test on-site since

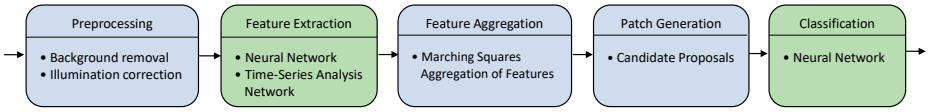


Fig. 2.9: Modification of the classic detection pipeline shown in Figure 2.8 for the use of neural networks. Template matching, time-series analysis, and the random forest classifier were replaced by a neural network. These replaced modules are colored green.

changes in the characteristics of artifacts have to be covered without the need for manual adjustments for each dataset. This demand leads to the employment of neural network approaches, which are presented in the following section.

2.2.4.2 Spatiotemporal Deep Learning

The detection of signals of interest from a real-world measurement always comes with irregularities. Changes in the environmental influence have to be taken into account, as well as the varying cleanliness of the samples. Dealing with these influences requires a technique that can adapt to the changes while concentrating on the typical characteristics of the particles of interest. At this point, the advantages of deep learning-based methods can be exploited. Instead of adjusting to a restricted scenario, a deep learning-based approach can take advantage of previous recordings by using them to approximate patterns that are typical for a particle of interest. Rather than calculating the features based on a static method, neural networks choose from a large set of possible feature extraction operations limited only by the number of freely learnable parameters and their architecture. Several deep learning approaches have been applied to the PAMONO recordings to evaluate their detection performance [385, 387, 733, 746].

Deep Learning Integration Into the Detection Pipeline There are different architectures of neural networks that can be integrated into the pipeline presented in Section 2.2.4.1. One pipeline modification using neural networks created to improve the flexibility of particle detection is presented in Figure 2.9. Several modules of the former pipeline shown in Figure 2.8 are replaced by neural networks for spatial or temporal classification of the respective inputs.

The key difference between the deep learning approach and the use of more direct methods like template matching is the way the parameters are used. By the layer-wise connection of learnable operators, this adaptation takes place on lower levels, such as edge detection, as well as on higher levels where possible particle shapes learned from training data can be correlated with a given image. For this purpose, backpropagation [264] is used in combination with a loss function, which in this case is the cross-entropy loss that is predicted in the current training step [387]. The whole process aims at the minimization of the evaluated loss functions, that is, the creation of a minimal divergence between predicted and expected classes.

The presented version with parts of the operations replaced by neural networks was able to achieve results on data with a median SNR value of 0.7 (see Equation 2.2) for images that were mainly affected by random noise. With the classic pipeline, this level was only possible on datasets with a median SNR value of 1.25. Although achieving slightly worse results compared with a template-matching approach which was optimized for each dataset separately, the neural networks can cover different situations without having to be adjusted anew [385, 733]. This property is highly desirable when handling shifts in the data characteristics, which are described in more detail in Section 2.2.6.2.

Deep Learning-Focused Pipeline The modified pipeline described in the previous paragraph increases the adaptability to changed characteristics of nanoparticle signals by replacing some classic operators with neural networks. At the same time, the modified structure still relies on inflexible methods for the extraction of candidate regions and patch extraction. The next step towards a completely adaptable structure is the employment of a neural network that is capable of proposing candidate regions by itself. In this way, different sizes of particle regions can be taken into account while learning the characteristic features of particles. When evaluating the architecture on datasets with different characteristics due to changing optical instruments and different image resolutions, accuracies of over 80 % could be achieved without adjusting parameters between analyses [733]. The pipeline which was proposed to achieve this goal is illustrated in Figure 2.10. It puts a stronger focus on the use of learned functionalities. The first step of detection with this approach is the spatial prediction of candidate regions in sliding window-preprocessed images. The downstream filtering of the candidates takes the temporal changes in pixel intensities into account and decides whether it corresponds to a characteristic of interest. Each of the two steps is learned using a neural network. The precise implementation of the spatial predictor is based on a Mask R-CNN [263], which has already been shown to be capable of handling the task of nuclei detection [302, 752, 761]. The Mask R-CNN itself uses a ResNet-50 Feature Pyramid Network [414] to generate abstract features for downstream detections. In terms of the classic pipeline shown in Figure 2.8, this functionality can be called a pattern detector. To speed up the training process and improve the generalization capability, we employ the concept of transfer learning and use the initial weights of a ResNet model pretrained on the Microsoft Common Objects in Context [415] dataset. Since the more universal, low-level patterns used for the detection process are already set, the training process can focus on adjusting the weights in the last layers containing high-level features. An additional advantage of the used network is that it is not bound to one specific tile size as in previous approaches. Rather, it can determine the sizes of particle regions from a given set of possible dimensions. This results in higher flexibility while retaining the possibility of restricting accepted sizes to a specific range based on external knowledge.

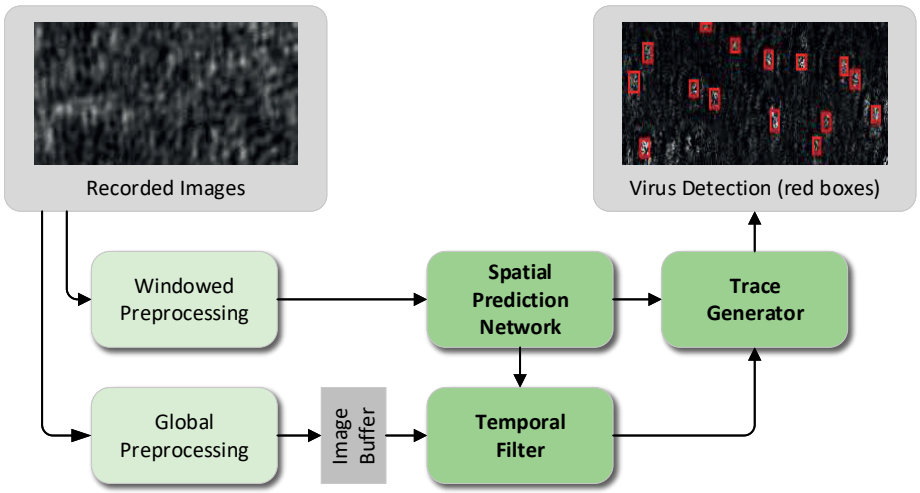


Fig. 2.10: Architecture of the spatiotemporal pipeline for particle detection. It preprocesses the recorded image stream in two ways, each fitting to its downstream task, predicts particle regions for each image, and connects these regions to countable particle traces stretching over multiple, consecutive frames [733].

The temporal filter network is kept simple, consisting of three fully connected layers with intermediate activation functions. It checks if the temporal view of the candidate region fits a particle signal and sorts out artifact signals that resemble the signals of interest spatially only for a short time. The still hand-crafted, fixed parts of the detection process can be found in two places: the preprocessing and the connection of confirmed particle regions to countable particle traces. By learning the core functionalities of the detection task, this approach was able to reach accuracies of over 80 % with respect to the expected particle counts of the test sets, which contained signals of 80 nm to 200 nm particles with different recording qualities, image sizes, and particle region sizes. The sets originate from different development stages of the PAMONO sensor, so setups with different optical instruments and camera configurations were included. The possibility of taking these differences into account indicates the high flexibility of the proposed solution.

In summary, classical methods such as template matching can work well for particle detection methods if they are specifically adapted to the specific imaging conditions. On the other hand, deep learning approaches provide better adaptability to distinct conditions with the drawback that they usually require a large amount of training data. However, specialized training approaches or the incorporation of domain knowledge can reduce the necessary training data. An example method dealing with this challenge explicitly is presented in Section 2.2.6.3.

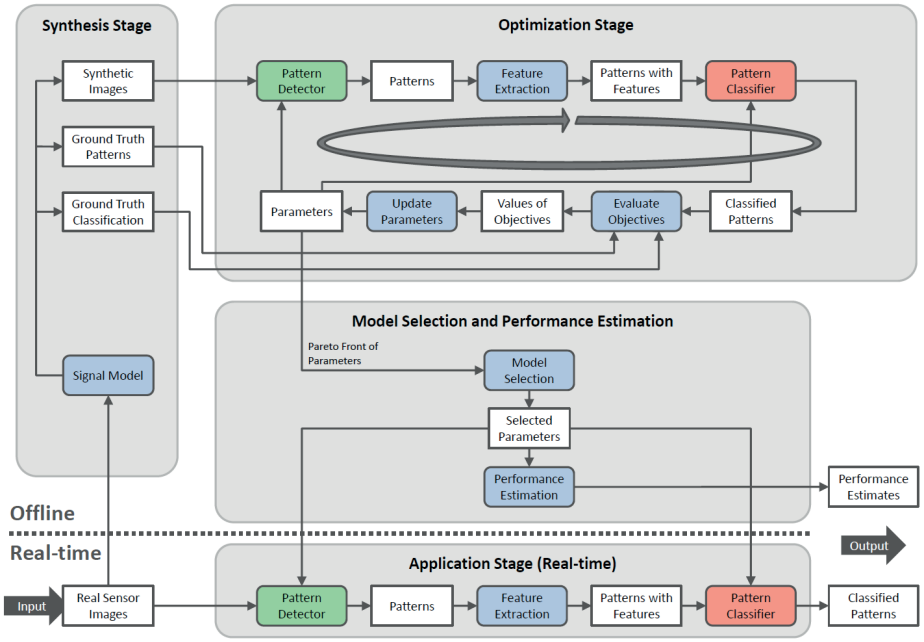


Fig. 2.11: The SynOpSis approach for automatic parameter adjustment. The optimization process tries to approximate the optimal parameters for the detection methods. Training is done on previously analyzed data. The amount of test data is increased artificially by a synthetic combination of background and particle signals of different training datasets. In the end, only the parameters of the detection method are changed without increasing the calculation complexity at test time [613].

2.2.5 Optimization

Most of the methods that can be used for object detection and other image-processing algorithms require suitable parameter values to be identified and set for improved detection. When trying to determine the best parameter settings for the detection of particles for a given algorithm or even for a complete pipeline as described in Section 2.2.4, the limit of what can be accomplished manually is reached quickly. A method utilizing a genetic optimization approach to handle this search automatically is presented in Section 2.2.5.1. Section 2.2.5.2 focuses on the possibilities of specialized optimization targeting an energy-efficient execution.

2.2.5.1 Algorithmic Optimization

A method that was developed for the purpose of automatic parameter selection is the SynOpSis (synthesis/ optimization/ analysis) approach [613], which is illustrated schematically in Figure 2.11. It makes use of previous examples and synthetic augmentations based on new combinations of particle signals and background signals from

different datasets to artificially increase the number of available images for training and testing. With a given interval of plausible values for the underlying parameters, it is run repeatedly with different parameter choices and with the goal of approximating the ideal settings for the given data. In particular, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [163], a multi-objective optimization approach, is exploited to find Pareto-optimal settings for the parameters used in the detection algorithms [613]. After that, the parameters that lead to the best results based on a predefined criterion are chosen for the single stages of the pipeline. While the optimization itself has to execute the detection pipeline after each optimization step, the actual detection is not slowed down. The reason for that is that after the optimization is done, the parameters are just transferred to the corresponding algorithms without conducting additional calculations while the actual detection takes place. Despite the benefits of tuning the algorithms to some training data, the optimization is limited to the given feature extractors and the parameter sets that are presented to it. A problem with this can appear when the selected operators fit a specific situation instead of generalizing, as can happen with template-matching approaches [385, 613].

2.2.5.2 Resource Optimization

The focus of resource optimization for the task at hand can vary depending on the concrete application scenario, while several goals can hinder each other in their fulfillment. Depending on the place of use, the energy efficiency of the used calculation platform can be highly important as a reliable external power supply can not be assumed everywhere. An energy-efficient device can thus cover a wider range of operating locations and provide better portability in general, as with lower energy consumption, smaller computing platforms like embedded systems can be operated for some time over by a battery. For this purpose, work on this subject could demonstrate a concept for balancing between a reduction of the overall energy consumption resulting in a higher battery lifetime and a shorter execution time [403]. The computations were either executed locally on a mobile graphics processing unit (GPU) or offloaded over a wireless network to transfer the sensor images as well as computation requests to a server. A complete offloading of computations was compared with an alternative in which the method can select the calculations that are offloaded based on a given objective such as energy-saving or a low execution time. For this purpose, a calculation of the consumed energy is required. This is achieved using the energy model

$$e_{\text{total}} = p_{\text{GPU}} \cdot t_{\text{GPU}} + p_{\text{GPU}} \cdot t_{\text{CPU}} + p_{\text{LTE}} \cdot t_{\text{LTE}}$$

which is based on the average powers p_{GPU}, p_{GPU}, and p_{LTE} of GPU, CPU, and the communication and the corresponding times t_{GPU}, t_{GPU} and t_{LTE} [403]. The result can be used to approximate the time that an algorithm can run with one charge. Although LTE was chosen as the communication technique, the energy model can be transferred

to other communication standards. To compute the overall runtime

$$t_{\text{total}} = t_{\text{GPU}} + t_{\text{CPU}} + t_{\text{LTE}} + t_{\text{Server}} - t_{\text{Parallel}}$$

the time t_{Server} needed by the server and the time t_{Parallel}, which can be saved by parallel operations on the client and the server, are also taken into account.

Using simulation software to simulate different hardware setups such as more powerful and more energy-efficient GPU alternatives, a share of 90 % of the energy could be saved compared with the configuration with the fastest calculation while optimizing, with the goal of a low overall execution time, a speedup of 55, that is to say, a division of the execution time by 55, could be achieved in comparison to the most energy-efficient configuration [403]. It was also observed that using the most power-efficient GPUs in the example setups did not lead to the most power-efficient total configuration. Instead, a GPU with more computing capacity could calculate results faster, leading to a lower total energy consumption [403]. Additionally, it was recognized that the usage of a GPU, in general, can reduce energy consumption significantly compared with an execution purely on a CPU [402].

2.2.6 Application in Real-World Scenarios

Besides the direct optimization of the detection methods to a specific data basis, there are further factors to consider in a real-world scenario.

2.2.6.1 Identification and Influence of Particle Size Distributions

While the most important task of the PAMONO sensor is to decide whether particles of interest are present in a sample, the determination of physical particle sizes brings additional advantages. A size distribution gives more information on the detected particles and enables plausibility checks helping to uncover outliers caused by impurities in the sample. Domain knowledge can be used for this purpose: since the device is adjusted beforehand to a specific particle type corresponding to the applied antibody coating, the expected range of particle sizes is known.

In tests with samples containing different, well-defined particle sizes, it could be demonstrated that the PAMONO sensor can be used to distinguish the different sizes from each other. The quality of the predicted size distributions was also compared with a commercial device showing that the predictions based on the PAMONO sensor rank at the same level of size prediction quality. For this purpose, the difference between the median signal intensity before and after a particle attaches is calculated. When comparing two particle sizes, this difference is proportional to the difference in signal intensities. An example of this is shown in Figure 2.6. A visualization of the analyses with the PAMONO sensor and, for comparison, a commercial nanoparticle tracking device for samples containing 100 nm, 200 nm, and 300 nm particles, as well as one mixture containing all three sizes, can be seen in Figure 2.12.

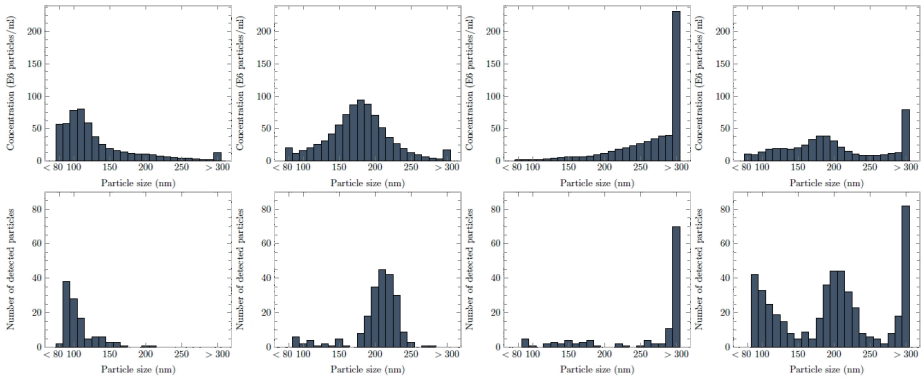


Fig. 2.12: Determined physical particle sizes for four different suspensions analyzed with (from left to right): 100 nm particles, 200 nm, 300 nm, and a composition of 100 nm and 200 nm and 300 nm particles. The top row shows reference distributions obtained with the commercial Malvern LM10 device while the bottom row shows the results of the PAMONO sensor [609].

The soft accuracy value of detected particle size distributions can be evaluated with

$$a^r = \frac{|\{e \in \mathcal{E} \big| |c^{\text{pred}}(e) - c^{\text{corr}}(e)| \leq r\}|}{|\mathcal{E}|}$$

calculating the share of predicted sizes c^{pred} that approximately match the correct size class c^{corr} for each patch $e \in \mathcal{E}$ of an analyzed image.

With the chosen division into classes of 10 nm, the classification of a 100 nm particle into say, the size class 80 nm, is tolerated with $r = 2$. At the same time, a prediction of 130 nm would be considered a false value for a 100 nm particle as a class interval of 10 nm together with $r = 2$ results in a tolerance of 20 nm. In an evaluation, particles of the sizes 80 nm, 100 nm, and 200 nm were analyzed. With classes of 10 nm intervals, an accuracy of over 70 % was reached for $r = 2$ [386].

In general, the PAMONO sensor can be calibrated by measuring the signal intensities of standardized particle sizes to map the measured values of temporal intensity steps to physical size. When changes are applied to the optical components of the sensor setup, the procedure has to be executed again as they can lead to a general shift in the measured signal intensities.

2.2.6.2 Detection of Viral Infections

The PAMONO sensor is designed to be usable as a rapid test for detecting viral infections on-site. When considered for use at an airport, a city center, or the entrance of a stadium, however, additional requirements emerge. For use as a rapid test on-site, the device has to be transportable to different places, which is fulfilled by the sensor case being suitcase-sized. Energy efficiency can be important for enabling the execution on an embedded device and use at otherwise poorly accessible places. A possible solution

for miniaturization was evaluated using the Hardkernel Odroid-XU3 and Odroid-XU4 single-board computers as a basis for calculations [399]. Although execution times are increasing, the ability to calculate and manage calculation offloading on embedded devices was demonstrated to be feasible [399]. The related method of computation offloading is described in Section 2.2.5.2.

2.2.6.3 Dealing With Imaging Artifacts

The amplification of imaging artifacts has to be expected when targeting a real-world application of the PAMONO sensor. Therefore, an essential aspect of reliable detection is the robustness of algorithms against imaging artifacts.

These artifacts originate from different sources. On the hardware side, there is the imperfection of the optical instruments and their adjustments. The smoothness of the gold film and its coating also influence the quality of the signals of interest. For example, scratches and other irregularities on the surface cause visible artifacts in the recordings. Another source that has to be considered is the limited and varying cleanliness of samples in real use cases. For example, air bubbles and dust particles in saliva or sputum samples cannot be avoided completely. In general, changing external influences have to be expected when targeting an application at places where the constancy of laboratory conditions cannot be achieved.

With variations of artifact types, such as line-like or wave-like structures, regions of constant intensities, pulsating regional patterns, or random noises, a detection has to tolerate both unstructured and structured artifacts.

This combination causes classic noise reduction methods to yield insufficient results. At the same time, changing patterns and intensities of artifacts also pose a problem for learning methods since many related approaches require a high amount of training data, especially when artifact characteristics differ in the recorded images.

Recent work tackles both problems by increasing the robustness of an existing learning approach relying only on a small amount of labeled training data.

This is achieved through a generative adversarial network (GAN) [237] that is not directly involved in the detection process but learns to simulate real artifact patterns to improve the detector before the actual detection takes place. Synthetically generated artifacts are used to overlay training images for the detection network. In this way, the detector learns how to tolerate the induced artifacts [547]. In addition, the architecture of the detection network itself does not need to be changed, so there is no decrease in the speed of the detection process.

The training of the GAN is not free from the demand of training data, but it uses reference images. These images are characterized by the fact that they do not contain particles of interest but only show imaging artifacts. The advantage of using this type of recordings instead of those carrying particle signals is that no physical test particles are required. This saves time and material costs and eliminates the risk of reproduc-

ing particle signals in the generated artifact images, as particles do not occur in the reference images. It is therefore ensured that only artifacts are learned.

Having demonstrated that it can produce realistic images with sufficient variation even with a small amount of training data, the StyleGAN2-ADA [312] was employed as the specific GAN architecture. While this GAN requires fixed-size training images and produces fixed-size images, different sensor configurations lead to different sizes of recorded images. Therefore, the GAN-based approach composes multiple patches of a fixed side length to cover the area of an annotated image for the training of the detector. As an illustration of the whole process, Figure 2.13 shows the generation of synthetic artifact signals and their usage in the training of the detection network.

Training a simple segmentation network with a downstream object detection shows the improvement in the robustness against artifacts when using the presented, GAN-based approach.

Two configurations were evaluated to compare the results with and without overlaying synthetic artifacts. The first configuration contained one dataset with particles of interest and only weak and unstructured artifacts. The second one uses the same dataset but adds synthetic artifacts generated by a GAN trained with reference images. Both employ the same test datasets, which contain different types and intensities of artifacts. After training two identical U-Net [544] architectures, each with one of the configurations, a clear difference becomes visible. While the training results without synthetic artifacts yield poor results, overlaying the images improved the mean F1 score by 22 % [547]. While improvements can be seen for all types of artifacts, they are most significant for images with structured artifacts of high intensities.

The GAN-based approach shows similarities to the approach presented in Section 2.2.5.1, as both create mixtures of particle signals and other signals from different datasets. The difference becomes clear when considering that the GAN-based generation does not directly use the limited set of natural images but artificially creates an arbitrary number of additional images in a learning process. This procedure can significantly increase the variability of the training images.

2.2.7 Current Research and Outlook

The techniques around the PAMONO sensor are expected to benefit from further research and development in the area of robust object detection. At the same time, it is worth evaluating an extension of the applications beyond virus detection, for which the sensor also forms a suitable basis. For each of the two perspectives, current and planned investigations are described below.

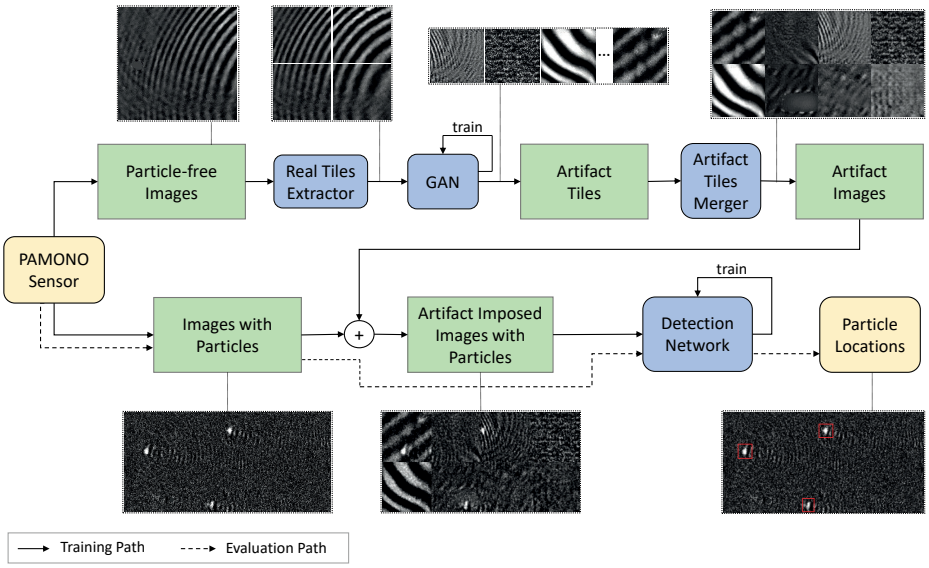


Fig. 2.13: Schematic representation of overlaying training images with generative adversarial network (GAN)-generated artifacts from composite tiles. The PAMONO sensor is used to record samples without particles of interest (upper part) and samples including such particles (lower part) for the training process. The trained detection model is then used to search for particles in images where their presence is unknown. Dashed arrows show the path of images in the evaluation process, while solid arrows represent the path of images in the training process. The images in dotted boxes visualize the single steps by examples. The yellow boxes illustrate the start and end of the pipeline, green boxes represent data, and blue boxes mark algorithms [547].

2.2.7.1 Improvements of Detection Capabilities

A method to improve the overall results concerns the physical setup of the sensor. With the rotation of the prism and the position of the camera objective influencing the signal quality, the appropriate adjustments must be realized before a recording. Although good results can be achieved by manual adjustments, there are limits to human accuracy when tuning the optical parts. For this reason, a sensor-actuator control system is targeted to determine an optimal setting automatically and to adjust the components in feedback with the resulting image signals.

In any case, further development in the direction of an on-site application should consider the need for high robustness of the methods against external disturbances to ensure a reliable and widely available rapid test. While the synthetic generation of artifacts presented in Section 2.2.6.3 showed improvements in the detection robustness, temporal dynamics are not specifically taken into account. By including temporal information, further improvements in the robustness can be assumed.

A different way of improving results is the incorporation of domain knowledge. The knowledge regarding a specific pathogen suggests the inclusion of particle counts per

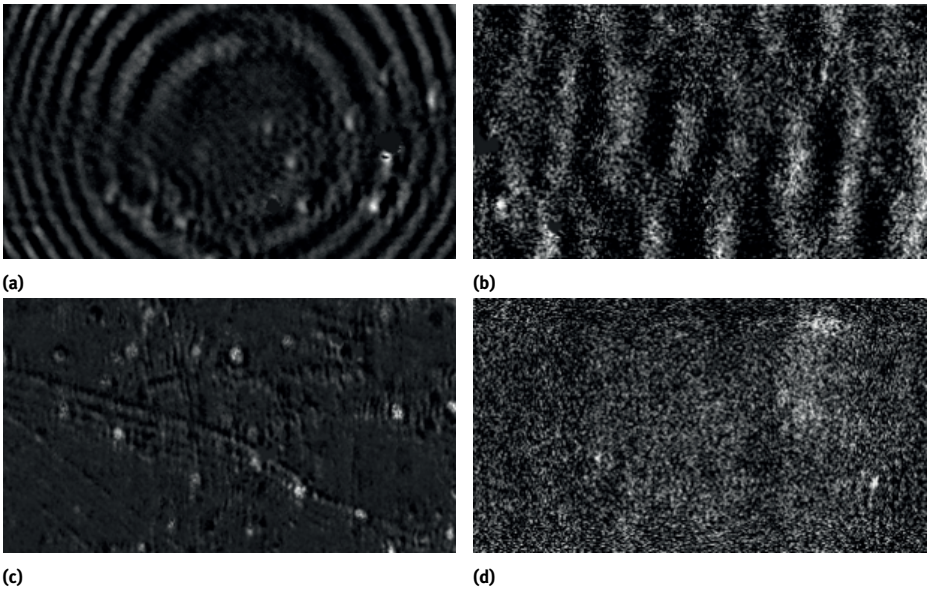


Fig. 2.14: Examples of different artifact characteristics in preprocessed images after the application of dynamic contrast enhancement.

image area as additional information. When a reference value for a usual concentration of those particles is known, the separation between samples with and without particles of interest can be improved based on an infection-specific concentration threshold. Although this option has not yet been quantitatively evaluated, it promises to improve reliability in practical applications.

A promising area of future exploration of mobile concepts for the execution of detection methods is the evaluation of embedded and other mobile devices. For example, the use of Field-Programmable Gate Arrays (FPGA), which are presented in Section 6.1 in Volume 1, promises to reduce power consumption while keeping the system size small. Another interesting work, which addresses hardware improvements, is the evaluation of learning approaches on modern memories in Section 7.2 in Volume 1. It focuses on the aspect of energy-saving in different approaches by targeting an optimal use of memory technologies.

2.2.7.2 Perspective Applications of the PAMONO Sensor

The PAMONO sensor proved its power in the characterization of viruses and Virus-Like Particles (VLPs). The latter is especially important since VLPs are often used as medical products and serve as a basis for different vaccines. Under these circumstances, the PAMONO sensor can be applied as an analytical instrument for quality controls during routine production of VLP-based vaccines, estimating the size and concentration of

produced NPs in real time. Further, the PAMONO sensor represents a valuable analytical instrument for the characterization of extracellular vesicles (EVs). In theory, the PAMONO sensor does not simply enable the sizing and quantification of EVs; it also helps to gain information about their surface markers and the molecules transported inside EVs. Nowadays, EVs earn growing interest as a means of intercellular communication. Acting this way, EVs can potentially serve as drug-delivering vesicles and as biomarkers of the cellular status. In this case, the ability of the PAMONO sensor to characterize EVs without labeling is a promising opportunity. Moreover, the PAMONO sensor may serve as a platform for the development of cell-based assays. In this case, the status of cells cultured on the sensor surface can be estimated via the cellular production of EVs and soluble mediators measured by the PAMONO sensor.

2.3 Cancer Diagnostics and Therapy from Molecular Data

Sven Rahmann
Alexander Schramm
Johannes Köster

Abstract: The past decade has seen unprecedented progress in the survival chances of cancer patients as a consequence of new treatments targeting tumor-specific cellular processes, which have been uncovered by molecular genetic analyses. From a data analysis perspective, the main challenge is the high dimensionality and multimodality of the genetic data relative to the small sample sizes (numbers of patients). From a computational perspective, the analysis of high volumes of data (about 100 GB of sequencing data for an individual tumor genome) currently requires high-powered computational resources and still remains challenging in the very short time frames that are desired to start treatment immediately.

We discuss two avenues of progress. First, we present methods that are able to extract most of the genetic variants from a sequenced tumor genome, but require only 2 % to 5 % of the computational resources compared with the current state-of-the-art procedures.

Second, we discuss a versatile unified statistical model for distinguishing true variants from technical artifacts of the DNA sequencing process.

Using analyses of paired samples from primary and relapse neuroblastoma tumors, we are able to extract patterns of tumor evolution that are correlated with cancer progression and the escape of tumors from therapeutic intervention.

As a result, a novel risk classification of neuroblastoma has been established based on genomic and mutational data.

2.3.1 Introduction

Cancer patients nowadays receive precise diagnosis and personalized therapy based on their individual molecular genetic data.

Here, we report on the analysis of DNA data from patients with neuroblastoma, a solid tumor typically occurring in children.

Diagnostics and prognoses are based on DNA sequencing, currently ranging from a few hundred targeted genes to entire genomes requiring 100 GB per patient in the near future.

Identifying relevant variants in the DNA that serve as biomarkers to distinguish between different risk classes, or primary tumors from relapses, or treatable versus non-treatable tumors, is and remains challenging, but every step of progress in this

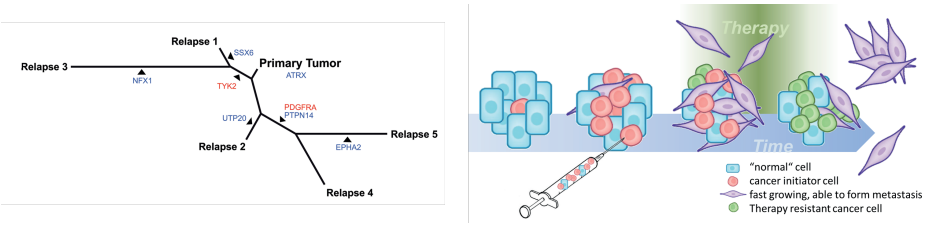


Fig. 2.15: Left: Evolutionary tree derived from binary features (presence/absence of informative single nucleotide variants) of a primary neuroblastoma tumor and several relapse samples taken from different tissues and at different times from the same patient [583]. Right: Illustration of evolution of tumor heterogeneity under therapy over time [588].

field helps to make better treatment decisions in the long run. The following molecular features derived from genomic data are of primary interest: (1) single nucleotide mutations (or variants, SNVs), (2) short insertions or deletions of DNA, (3) large structural variants (e.g., chromosomal translocations), (4) copy number changes (gain or loss of genetic material in tumor cells), (5) epigenetic changes, such as DNA methylation changes, and (6) differences in gene expression.

Over the past years, we have developed feature extraction workflows and data analysis processes for each type of feature mentioned above. For data analysis workflows in general, but especially in medicine, reproducibility of derived data from raw data is of utmost importance. The basis of each of these processes is our workflow management system called Snakemake [345, 450], which is now widely used worldwide, as it guarantees reproducibility in particular for large-scale DNA sequence analysis workflows. Furthermore, the Bioconda package repository [242] was founded by one of us (JK) and now, with widespread community support, acts as a central repository for semantically versioned bioinformatics software, which is made available in a reproducible way.

In the following, for simplicity, we focus on the first type of features (SNVs), but these findings also translate to the other variant types, if appropriately adjusted. In particular, we discuss whether we can determine genomic variants that distinguish primary neuroblastomas from those that re-occur after therapy (referred to as relapses or relapse samples). The latter are responsible for adverse disease courses and are currently considered to be incurable. It was therefore highly encouraging that we were able to identify several genes with recurrent mutations present only in relapse samples [583]. Figure 2.15 summarizes some of our key findings on tumor heterogeneity after relapse (left side) and illustrates the tumor evolution process (right side). It is mainly this developing molecular heterogeneity of tumor cells under treatment that currently prohibits effective long-term therapies.

The main resource constraints for this setting and similar situations are a limited number n of samples (patients) versus an extremely high number p of potential features (e.g., each potential variant in the genome observed in at least one sample).
So we face two challenges in particular:

1. resource-efficient detection of candidates of variants (Section 2.3.2)
2. accurate classification of candidates in each sample (true variant vs. noise, technical artefact, etc.; Section 2.3.3)

2.3.2 Resource-Efficient Detection of Variant Candidates

Standard genetic mutation or variant analysis starts with an extremely compute-intensive step: the localization of every single sequenced DNA fragment (or "read"; there are literally millions of DNA reads in a single dataset) in the genome, and a pairwise comparison between the fragment and the genomic sequence. Such pairwise alignments are the basis of variant calling: many reads showing a certain difference at the same position compared with the reference genome, this provides convincing evidence that the sequenced genome contains a specific genetic variant at that position, either in both inherited chromosome copies (homozygous variant) or in just one (heterogzygous variant). To be precise, complex statistical models and tests are necessary to distinguish true variants from possible technical artifacts (see Section 2.3.3).

This first localization and comparison step is performed by so-called *read mappers*, such as BWA-mem [391], bowtie2 [371], minimap2 [390], or PEANUT [344]. Extensive parallelism on both multi-core systems and GPUs keep the (wall clock) time of this step within a few hours. However, the overall CPU work consists of many CPU days or months for a single dataset, consuming considerable energy.

It is therefore of high interest to develop more resource-frugal methods to achieve the same task, or at least a large fraction of it. We explored *alignment-free* methods as an alternative to the above mapping and alignment-based method. In particular, we propose to use short DNA strings of length k (so-called k-mers) to directly detect potential single-nucleotide variants, as we now describe.

2.3.2.1 Genome Preprocessing

We first preprocess the reference genome.

1. Select an appropriate value for k, such that most k-mers are unique in the reference genome. Our studies indicate that $21 \leq k \leq 31$ works well for the human genome [517].
2. Build a (very large) hash table of k-mers in the human reference genome and the number of times that they occur. We need to take into account that double-stranded DNA is equivalent to its reverse complement.
3. Mark the unique k-mers; they point to a unique position in the genome.
4. Among the unique k-mers, mark those that are *robustly unique* against single substitutions, i.e., those for which no Hamming-distance-1 neighbor also occurs in the genome. The resulting robustly unique k-mers do not only point to a unique

location in the genome; they also cannot easily be changed into *k*-mers that occur at a different genomic location.

The alignment-free methods work either exclusively on the robustly unique *k*-mers or on all unique *k*-mers, giving the information from the robustly unique *k*-mers a higher weight. This pre-processing step has to be performed only once for any genome version.

We found that multi-way bucketed Cuckoo hash tables are ideally suited for the task, as they allow relatively quick construction times and yield very fast lookup times later. There are smooth trade-off options between lower memory usage and even faster lookup times.

In a preliminary study [756], we designed and implemented these hash tables for a simpler application than variant calling: xenograft sorting. Here, a human tumor is engrafted into another organism (typically a mouse) to be able to study its evolution and response to different therapies. When such a tumor is sequenced, one obtains a mixture between human and mouse DNA reads, so that all reads have to be assigned to the organism of origin before proceeding further. This assignment is called xenograft sorting. Even though human and mouse are quite similar on a genetic level, they can be sufficiently well distinguished on the *k*-mer level. We presented a classification method based on *k*-mer hash tables, as outlined above, with extremely high accuracy, but using much less CPU work than previous methods: less than 25 % of comparable hash-based methods and less than 5 % of classical alignment-based methods [756]. We additionally showed that the placement of keys in the hash table can be optimized to yield optimal average look-up times (based on the number of random memory accesses, i.e., likely cache misses), saving 10 % to 15 % of CPU work for each sample (after a 48 CPU hour optimization procedure that has to be run only once) [757].

2.3.2.2 Basic Alignment-Free Variant Calling

The underlying idea of this method is as follows: We count all the *k*-mers in a sequenced sample and produce a histogram of the count values. A typical (unique) *k*-mer should have a copy number of two (in a diploid genome) when no variant is present. We therefore analyze the histogram of observed *k*-mer counts (Figure 2.16) from the sample. The leftmost peak (counts near zero) can be explained by rare *k*-mers due to sequencing errors or contamination; we can attempt to correct these, or ignore them entirely. We fit a negative-binomial mixture model to the remaining peaks occurring at equidistant counts. The main peak corresponds to a copy number of 2 in a diploid genome (from *k*-mers present in both the maternal and paternal chromosome set).

The initial analysis is restricted to the robustly unique *k*-mers from the reference genome. We expect that each such *k*-mer has a copy number of either 0 (homozygous variant), 1 (heterozygous variant) or 2 (no variant) in the sample. Higher copy numbers could be explained by segmental duplications, which we do not consider at this point. If we suspect a variant, we look for *isolated single nucleotide variants*, i.e., *k*-mers with

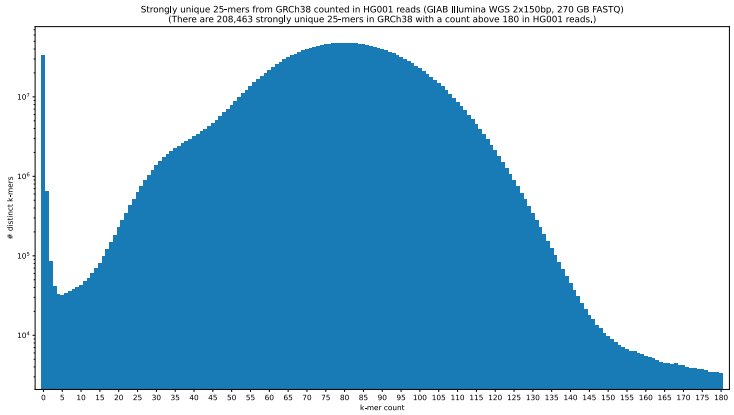


Fig. 2.16: Illustration of a *k*-mer count histogram, relating observed *k*-mer counts (x-axis) to their frequency (y-axis; logarithmic). The leftmost peak (close to zero) represents noise and erroneous *k*-mers, mostly due to sequencing errors. The main peak (near 80, approximately the sequencing coverage of this example) corresponds to the standard copy number of 2. The shoulder (near 40) then corresponds to a copy number of 1 and consists of *k*-mers that are part of heterozygous isolated mutations. This histogram was created from a control sample; in a tumor sample, more irregularities, especially additional peaks at higher copy numbers, can be expected.

a Hamming distance of 1 to the reference *k*-mer, among the *k*-mers in the sample. If we find a unique one (with the expected copy number), we store the pair of reference *k*-mer and modified *k*-mer as a candidate for a variant.

This process can be implemented very efficiently, and in addition, it can be trivially parallelized. It yields candidates for Single Nucleotide Variants (SNVs) that then can be checked by statistical methods (see next section). It can also reliably detect copy number variants on long segments. However, it cannot easily detect more complex variants, such as two SNVs in close proximity, short indels, or structural variants: Here translating *k*-mer information into an exact variant is more difficult, but can resort to alignment-based methods for the local regions around areas with suspicious *k*-mer frequency structure.

Perspectives Alignment-free variant calling is still an active research area, and while we made contributions to the underlying data structures (engineered Cuckoo hash tables) and were successful in calling selected SNVs, further ideas are necessary to call larger classes of variants reliably. Possible approaches include using locality sensitive hashing, in particular min-hashing, instead of exact *k*-mer hashing, combined with *hybrid methods* between alignment-free and alignment-based approaches. To assess the potential of min-hashing-based methods, we conducted a detailed statistical feasibility study, examining when it is useful to include known variants into a *k*-mer-based read mapper (and when not; see [515]), paving the way for novel approaches.

2.3.3 A Unified Statistical Model for Genomic Variant vs. Artifact Classification

We present an extension and generalization of a latent variable model originally published by Köster, Dijkstra, Marschall, and Schönhuth [342].

For this, we consider a set S of samples. Samples can be related with each other in three ways.

1. There can be clonal inheritance between two samples $s_1, s_2 \in S$: sample s_1 inherits all constitutive genetic variants of sample s_2. In addition, the tissues of origin of both samples may have developed their own *somatic* mutations during their lifetime until sequencing.
2. There can be Mendelian inheritance [443] between samples $s_1, s_2, s_3 \in S$: the individual of origin of sample s_1 inherits constitutive genetic variants of two parental individuals (s_2 and s_3).
3. A sample $s \in S$ can be contaminated with a fraction of another sample $s' \in S$.

We represent the three relationships in a directed graph $G = (S, I_c, I_m, C)$ (the *sample graph*) with edge types $I_x \subseteq S \times S$ for clonal ($x = c$) and Mendelian ($x = m$) inheritance as well as $C \subseteq S \times S$ for contamination. The corresponding contamination fraction can be obtained with $c : C \to [0, 1]$.

The above representation can be used to model the three classical cases of genomic variant calling: single-sample or population calling (the graph has no edges) [171], pedigree based family variant calling (Mendelian inheritance edges) [171], and calling of tumor/normal sample combinations (clonal inheritance and contamination edges) [342]. Importantly though, instead of being limited to these, it can reach beyond them by combining the mechanisms into more complex scenarios.

2.3.3.1 Variables and Notation

Observed Variables For each potential genomic variant of interest, we observe sequencing read data $\boldsymbol{Z}_s = (Z_1^s, \ldots, Z_k^s)$. If the read data consists of so-called paired-end reads (each investigated DNA fragment is sequenced from both ends), each observation in $Z_i^s \in \boldsymbol{Z}_s$ is a tuple $Z_i^s \in (\{\text{A,C,G,T}\}^+, \{\text{A,C,G,T}\}^+, \mathbb{N})$, with the first and the second element denoting the nucleotide sequence of the read and the last element denoting the so-called observed insert size, that is, the number of bases from the leftmost to the rightmost covered base when aligning the read pair to the most likely position of origin on the reference genome of the investigated species. If the read data consists of so-called single-end reads (each investigated DNA fragment is sequenced just from one end), each observation $Z_i^s \in \boldsymbol{Z}$ is simply the nucleotide sequence of the read, in other words $Z_i^s \in \{\text{A,C,G,T}\}^+$.

Latent Variables The central readout of our model is the allele frequency in each sample s, denoted as latent variable $\theta_s \in [0, 1]$. For each read observation i, there is

a binary latent variable ξ_i^s with $\xi_i^s = 1$ denoting that the observation originates from the variant allele (i.e., from a genome copy hosting the variant under consideration) and $\xi_i^s = 0$ denoting that the read originates from the reference allele (i.e., from a genome copy hosting exactly the same sequence as in the reference genome of the corresponding species). In addition, a binary latent variable ω_i^s, denoting whether the observation has been aligned to the correct ($\omega_i^s = 1$) location of origin in the reference genome, is used.

Extensions for Bias Estimation The model can be further extended in order to estimate biases in additionally observed properties of the read data, that is, the strand, the read position supporting the variant, the read orientation, and whether the alignment against the reference genome covers the entire read. Biases from an equal distribution in the observed values of variant supporting reads for any of these properties typically indicate an artifact. For clarity and brevity, we omit the integration of these biases in our model here. An integration of strand bias can be already found in [342].

2.3.3.2 Latent Variable Model

In the following, we briefly introduce the latent variable model used for calculating allele frequency likelihoods that has been published recently [342], and then provide a generalization of the method. When evaluating if a read deviates from the reference genome, two types of uncertainty are to be considered. First, there is *alignment uncertainty*: often, a read can be aligned at multiple loci in the reference genome (also see Section 2.3.2).

Depending on their similarity, there is more or less certainty about the optimal positioning of the read. Read mappers and alignment tools, such as BWA [391], report this uncertainty as mapping quality (MAPQ), which can be translated into a probability π_i^s associated with each read observation Z_i^s to be aligned to the correct locus. Second, there is *typing uncertainty*: the observed read sequence is not a perfect representation of the true DNA fragment that has been sequenced, but instead a measurement entailing potential errors and artifacts. The DNA sequencing machine provides an estimate of the certainty of each reported base as the so-called base quality, which can again be translated into a probability of the reported base to be correct. In addition, depending on the sequencing technology, there are known rates of false insertions or deletions of bases in the reported read sequences, as remarked on for example by Schirmer, D'Amore, Ijaz, Hall, and Quince [578].

We now model the relationships between our observed and latent variables, while taking above mentioned uncertainties into account. For each observation Z_i^s in sample s, we handle alignment uncertainty by defining the distribution of the latent variable ω_i as

$$\omega_i^s \sim \text{Bernoulli}(\pi_i^s). \tag{2.3}$$

The distribution of the latent variable ξ_i^s depends on the expected fraction of observations from the variant allele. If s is not contaminated by another sample, we define

$$\xi_i^s \sim \text{Bernoulli}(\theta_s \tau). \tag{2.4}$$

Thereby, $\tau \in [0, 1]$ denotes a sampling bias that occurs because it is usually harder to obtain observations from the variant allele: it is harder to align, and depending on the size of the variant, harder to obtain reads that sufficiently cover it [342]. If, in contrast, s is contaminated by a s' (i.e., $e = (s, s') \in C$) with fraction $\alpha = c(e)$ we define

$$\xi_i^s \sim \text{Bernoulli}(\alpha\theta_s \tau + (1 - \alpha)\theta_{s'} \tau). \tag{2.5}$$

In other words, the expected fraction of observations from the variant allele becomes a mixture of the allele frequencies in s and s'.

Then, typing uncertainty can be modeled as

$$Z_i^s \mid \xi_i^s, \omega_i^s \sim \begin{cases} p_i \text{ if } \xi_i^s = 1, \omega_i^s = 1 \\ a_i \text{ if } \xi_i^s = 0, \omega_i^s = 1 \\ o_i \text{ if } \xi_i^s = 0, \omega_i^s = 0. \end{cases} \tag{2.6}$$

Here, a_i, p_i, and o_i are probability distributions modeling the case that the observation comes from a genome copy where the variant is present (p_i), absent (a_i), or from a different locus (o_i). These can be computed using Pair Hidden Markov models, which essentially realign the read sequence against the sequence of reference and alternative allele while statistically considering sequencing error rates, as shown in Köster, Dijkstra, Marschall, and Schönhuth [342] for deletions and insertions. Since then, via analogous approaches, our model has been extended to also support all other common variant types ranging from small (SNV, MNV) to structural variants such as inversions, duplications, and arbitrary chains of breakpoints.

By combining the above relations, the model can be used to calculate the likelihood of a given combination of allele frequencies of samples $S = \{s_1, \ldots, s_n\}$ as

$$\Pr(\boldsymbol{Z}_{s_1}, \ldots, \boldsymbol{Z}_{s_n} \mid \theta_{s_1}, \ldots, \theta_{s_n}) = \prod_{j=1}^{n} \prod_{i=1}^{|\boldsymbol{Z}_{s_j}|} \Pr(Z_i^{s_j} \mid \theta_{s_1}, \ldots, \theta_{s_n}) \tag{2.7}$$

while assuming independence between the read observations. Note that the computation of the likelihood function is linear in the total number of read observations, as we have shown previously [342].

2.3.3.3 Prior Distribution

The prior probability of a given allele frequency combination $\theta_{s_1}, \ldots, \theta_{s_n}$ in our generalized model can be computed by considering the dependencies between the samples modeled by the sample graph G (see beginning of Section 2.3.3). In addition, we assume

that for each sample $s \in S$, a ploidy $\rho_s \in \mathbb{N}$ (which may differ by chromosome, e.g., it may be sex-specific), a somatic effective mutation rate $\mu_s \in [0, 1]$, and a germline mutation rate $\nu_s \in [0, 1]$ are known. For calculating a prior probability, the key is to explain the total allele frequency θ_s by a germline allele frequency ι_s and a somatic allele frequency $|\theta_s - \iota_s|$. Usually, one of the two will be zero, such that variants are explained either by germline or by somatic mutation, but combinations thereof can also happen in rare cases. From the known ploidy ρ_s of a sample $s_j \in S$, we can calculate the set of possible germline allele frequencies $\zeta_s \subseteq [0, 1]^{\rho_s+1}$. For example, for ρ_s, we obtain $\zeta_s = \{0, 0.5, 1\}$; in other words, any germline variants may occur either in no, one allele (0.5 or 50 %), or two alleles (1.0 or 100 %). The prior probability can then be calculated by recursively exploring all possible explanations of a given total allele frequency combination.

For a combination of germline and somatic allele frequencies we can then distinguish between the following cases:

1. All samples that are not direct descendants of other samples (have no incoming edges in I_c and I_m in the graph G) are considered a population and the prior probability of their combination of germline allele frequencies is calculated, as defined by DePristo et al. [171], based on a so-called heterozygosity (i.e., the expected proportion of heterozygous sites in the genome), which is usually known for the investigated species.
2. For any sample $s \in S$ that inherits clonally from another sample $s' \in S$, we calculate the prior probability for the somatic allele frequency $f = |\theta_s - \iota_{s'}|$ according to the method of Williams, Werner, Barnes, Graham, and Sottoriva [730], who report a formula for the expected cumulative number of somatic mutations per frequency. The latter can be translated into the corresponding density by normalizing with the genome size g and taking the first derivative, resulting in $h(f) = \frac{\mu}{f^2 \cdot g}$ for $f > 0$. In order to also be able to calculate the probability for $f = 0$, we define a reasonably small ϵ and define $h(0) = 1 - \int_{\epsilon}^{1} h(f)\mathrm{d}f$.
3. For any sample $s \in S$ that inherits in a Mendelian [443] way from two parents $s' \in S$ and $s'' \in S$, we first calculate the number of expected constitutive alternative alleles in the child and the parents by multiplying the ploidy with the respective germline allele frequency, i.e., $\rho_s \cdot \iota_s$. We then sum over the probabilities of all cases of inheriting chromosomes with or without the variant allele from the parent samples that could explain the expected constitutive alternative alleles. The individual probabilities can be calculated by modeling an urn drawing process without replacement, yielding a hypergeometric distribution. Finally, additional somatic variation, i.e., cases where $\theta_{s_j} - \iota_{s_j} \neq 0$, are handled by multiplying the corresponding prior probability for the somatic allele frequency.
4. Finally, sometimes it might not be possible to formulate prior assumptions about allele frequencies of a sample $s \in S$. In such cases we specify an allele frequency universe $U_s \subseteq [0, 1]$ for a sample and assume a uniform distribution.

By taking the product over the priors for individual or groups of samples derived from distinguishing the above three cases, the prior probability for any combination of germline and somatic allele frequencies can be obtained.

2.3.3.4 Variant Calling Grammar

The above model is implemented in the software Varlociraptor (https://varlociraptor.github.io). Varlociraptor offers a *variant calling grammar* that allows to define a scenario that configures all aspects of the model (prior parameters, sample graph) via a textual representation in YAML format (YAML Ain't Markup Language; https://yaml.org/). A scenario consists of the following sections.

Species In this section, general prior knowledge about the investigated species is defined, such as the heterozygosity (see Section 2.3.3.3) and the ploidy (number of chromosome copies in a cell). The latter may be defined with sex-specific exceptions (such as the X and Y chromosome distribution in humans).

Samples In this section, the samples and their dependencies (i.e., the sample graph) are defined. For each sample, it is necessary to either define an allele frequency universe (leading to a uniform prior across the defined frequencies) or the sex. In the latter case, ploidy and heterozygosity are taken from the species definition and used to configure the prior accordingly. Each sample may be annotated with a contamination by another sample in a given fraction (this can be used to define the common case of having a tumor sample that also contains healthy normal tissue). Finally, each sample may define a type of inheritance (Mendelian or clonal), while referring to the corresponding parental samples.

Events The heart of a scenario is formed by the definition of mutational *events* of interest. These can be used to define any kind of Boolean logic expressions over allele frequencies (discrete or intervals) in the given samples.

An example for a scenario modeling the calling of variants in a patient for which a normal healthy blood sample, a tumor sample, and a relapse sample is used can be seen in Figure 2.17. Here, for simplicity, we have initially not defined any prior knowledge regarding mutation rates etc., thereby modeling a uniform distribution over the defined allele frequency universes. An equivalent scenario including this kind of prior knowledge is shown in Figure 2.18. Here, it can be seen that we are able to define inheritance between the normal and the tumor sample. For the relapse sample, although in principle it should inherit mutations from the tumor sample, it is unknown to what extent this happens, because usually only one or a few subclones survive the therapy. Hence, we refrain from specifying an inheritance between the tumor and the relapse, and instead impose a uniform prior on the possible allele frequencies in the relapse sample.

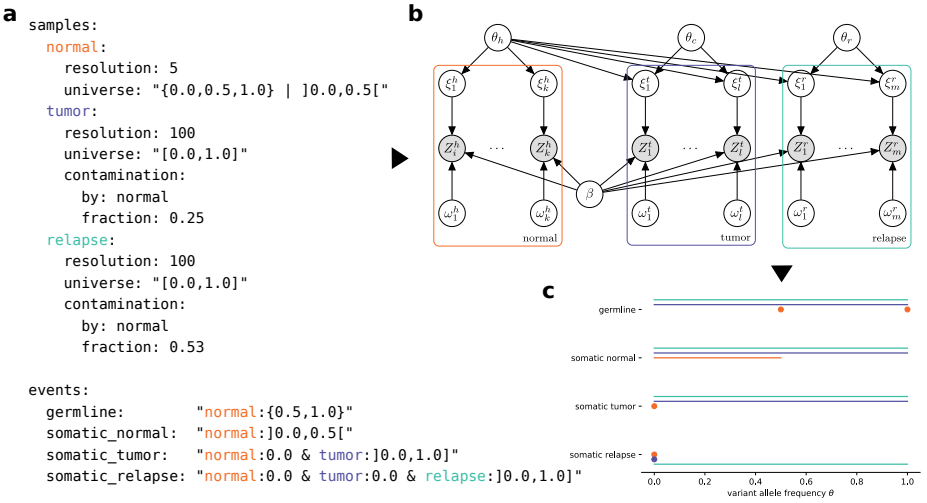


Fig. 2.17: Example of a Varlociraptor scenario specification to distinguish between germline variants and those occurring as somatic events in the primary or relapse sample. (a) Scenario definition via Varlociraptors variant calling grammar. The first section defines the three involved samples normal healthy blood, primary tumor, and relapse after therapy, along with their contaminations and expected allele frequency universe. The second section defines the events of interest via Boolean logic formulas. (b) The resulting structure of the latent variable model, automatically derived from the scenario definition. (c) Visualization of the expected allele frequencies in the three samples for each defined event.

2.3.4 Application and Results

It was previously shown that Varlociraptor is able to significantly improve the recall, while precisely controlling the false discovery rate without the need to tune any technical filter parameters in the absence of a biological interpretation [342]. Here, we illustrate the application of the model by re-analyzing the aforementioned previously published neuroblastoma dataset [583]. In this manuscript, we analyzed genomic data from 17 neuroblastomas, for which DNA was available from the primary tumor and the tumor at relapse. Obtaining the sequence of the entire coding region of the human genome (usually referred to as the "exome") was especially useful for modeling intra-tumor heterogeneity and clonal tumor evolution.

We use the normal-tumor-relapse model formulation from Figure 2.18 and parametrize it as follows. The effective somatic mutation rate in the tumor sample is set to $2.93 \cdot 10^{-6}$. This roughly models the expectation of at most 100 de-novo somatic mutations in typical neuroblastoma tumors found in our original study [583].

Since somatic mutation can also appear in the normal tissue, we set the corresponding effective somatic mutation rate to $2.8 \cdot 10^{-7}$, as reported by Oota [485]. Finally, the

```
species:
  heterozygosity: 0.001
  genome-size: 3.1e9
  ploidy:
    female:
      all: 2
      X: 2
      Y: 0
    male:
      all: 2
      X: 1
      Y: 1

samples:
  normal:
    sex: female
    somatic-effective-mutation-rate: 2.8e-7
  tumor:
    sex: female
    somatic-effective-mutation-rate: 2.93e-6
    inheritance:
      clonal:
        from: normal
    contamination:
      by: normal
      fraction: 0.1
  relapse:
    resolution: 100
    universe: "[0.0,1.0]"
    contamination:
      by: normal
      fraction: 0.53

events:
  germline:        "normal:{0.5,1.0}"
  somatic_normal:  "normal:]0.0,0.5["
  somatic_tumor:   "normal:0.0 & tumor:]0.0,1.0]"
  somatic_relapse: "normal:0.0 & tumor:0.0 & relapse:]0.0,1.0]"
```

Fig. 2.18: Extension of the Varlociraptor scenario specification in Figure 2.17 to include prior knowledge. We define the species (here *Homo sapiens*) in terms of genome size, heterozygosity (expected fraction of heterozygous loci), and sex-specific ploidy (number of chromosome copies). In addition, we model known somatic mutation rates, and define that the tumor inherits germline mutations from the normal sample.

tumor and the relapse sample tissue is usually contaminated by healthy cells. We use the amounts of contamination reported in the original study [583].

Workflow Analyzing sequencing data for genomic variants entails a variety of steps, which we outline in Figure 2.19. The entire analysis is implemented as a Snakemake workflow [343].

First, raw reads are processed by (a) trimming so-called sequencing adapters, (b) mapping them to the reference genome of the corresponding species, (c) removing putative duplicates from the Polymerase Chain Reaction (PCR), and (d) recalibrating base qualities. Sequencing adapters (a) are non-biological artifacts of the sequencing process. Since they are known beforehand, they can be removed from the reads by

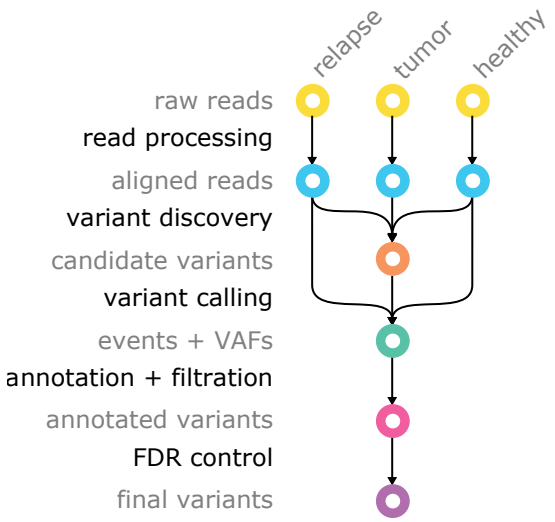


Fig. 2.19: Schematic representation of applied genomic variant calling workflow. Nodes represent original or derived data (gray labels in the left column), arrows represent processing steps (black labels in the left column).

performing an error-tolerant alignment between each read and the known sequence. We use Cutadapt to perform this step [431]. By mapping reads to the reference genome (b), we obtain the correct order and individual differences of each read compared with the representative genome of the underlying species. The resulting read alignments already contain all necessary observations for applying the Varlociraptor model. In order to obtain a signal of sufficient strength, sequencing protocols often entail the amplification of the DNA material via polymerase chain reaction [24]. The result is that there can be multiple reads from the same DNA fragment. Since Varlociraptor assumes each read to be an independent observation, it is important to remove such putative PCR duplicates, which we achieved using Picard tools [500]. Finally, the sequencing process sometimes causes artifacts to appear next to certain motifs [19]. In (d), we therefore use the base recalibration process from the Genome Analysis ToolKit (GATK [171]), which systematically investigates base alteration causing motifs and recalibrates the per base confidence scores in each sequencing read to reflect the uncertainty about whether an altered base is a true signal or a motif-induced artifact.

Second, the aligned reads are used to generate candidate variants. We use the tools Freebayes [222] and Delly [523] for this purpose. While the former covers small variants that can be covered by a single read (SNVs, MNVs, small insertions and deletions), the latter covers large, structural variants (large insertions and deletion, inversion, and duplications). Importantly, while both Freebayes and Delly provide their own statistical models for calling variants, we utilize them to generate candidate hypotheses. Both

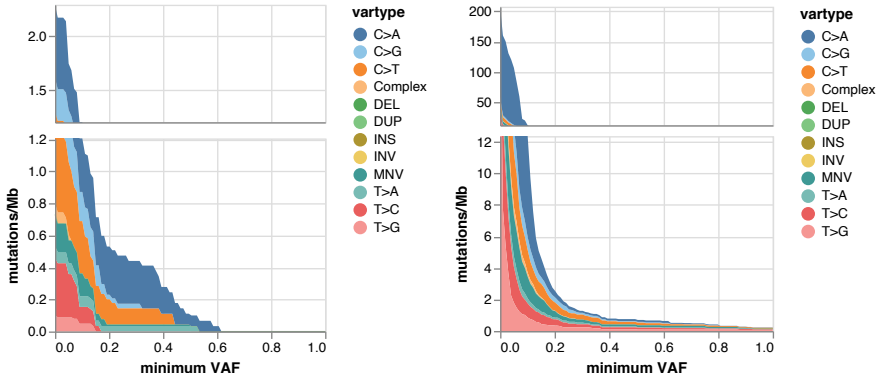


Fig. 2.20: Mutational burden of patient 1 from Schramm et al. [583] in the primary tumor (left) and relapse sample (right). The horizontal axis shows the minimum allele frequency, vertical axis shows the mutational burden as number of coding somatic mutations (calculated as expected value over the posterior probability for having a somatic mutation) per megabase of coding genome. The colors represent different types of mutations (see legend).

models are designed only for specific cases and are not generic enough to handle the composition of samples available in this dataset.

Third, we use Varlociraptor to (a) extract observations for each sample and each candidate variant and (b) apply the model as defined in the corresponding scenario for each patient in the study data.

Fourth, we (a) annotate the variant calls from Varlociraptor with their impact on proteins via the VEP tool [440] and (b) filter them for those that are of interest. In this case, we strive for three disjoint sets of variants

1. Variants that have been previously described as pathogenic or likely pathogenic in other studies.
2. Variants with high impact on the protein but which have not been previously described by other studies.
3. Variants with moderate impact on the protein but which have not been previously described by other studies.

Finally, we separately control the local false discovery rate for somatic variants in either the tumor or the relapse sample on each of the three sets.

Insights In the following, we summarize the most important insights from reanalyzing the study data with this workflow.

Figure 2.20 shows the mutational burden as a curve over the minimum allele frequency on an example patient. It can be seen that the burden for higher frequencies in general increases in the relapse sample compared with the tumor sample. This supports the hypothesis that the relapse sample originates from a subclone of the tumor

sample, which has survived therapy. Thus, one can expect that resistance-inducing mutations in the relapse sample become more abundant. Our findings contribute to the emerging view of resistance to cancer therapies as an evolutionary process. Selection of surviving clones results in mutational fingerprints that are specific for resistant or recurrent tumors. A better understanding of these genetic fingerprints is a prerequisite for identifying markers allowing early detection of resistance or tumor recurrence and enabling timely adjustment of therapies to further improve the survival and cure of cancer patients.

Future work entails the interpretation of individual recurrent deleterious gene and pathway alterations across the analyzed samples. Moreover, we aim to further improve the prior model of Varlociraptor such that assumptions about subclonal inheritance patterns can be incorporated as well.

Finally, we will combine the statistical approach of Varlociraptor with alignment free methods, as outlined in Section 2.3.2. Since Varlociraptor has to perform a realignment of read sequences anyway (see Section 2.3.3.2), we may replace the initial read alignment with an alignment free approach that yields a rough positioning of reads on the reference genome so that they can be selected for validating a given candidate variant with Varlociraptor. For this, it is necessary to accurately estimate the alignment uncertainty from the k-mer hits via, say, the strategy proposed in our previous work on PEANUT [344]. Finally, the detection of candidate variants with alignment free methods has to be extended beyond single nucleotide variants. Here, a possible strategy might be a hybrid approach where aberrations in k-mer counts are translated into an exact variant call by (a) collecting the causing reads, (b) assembling them into one or more consensus sequences [114], and (c) aligning these against the reference genome to determine the nature of the variant.

2.4 Bayesian Analysis for Dimensionality and Complexity Reduction

Zeyu Ding
Katja Ickstadt
Alexander Munteanu

Abstract: In this contribution, we will present Bayesian approaches for dimensionality and complexity reduction in the context of health-related problems. Following an introduction to Bayesian analysis in general, we will first show two examples of Bayesian variable selection methods for reducing the number of variables, one for binary data with an application to Single Nucleotide Polymorphisms (SNPs) for the HapMap dataset, and one for time-to-event endpoints with an application to glioblastoma data from the Cancer Genome Atlas. Second, we will present an approach for reducing statistical models, where we transfer the Merge & Reduce principle to maintain statistical summaries in streaming models. The variable selection approaches as well as the Merge & Reduce approach are important steps towards resource-aware data analyses.

2.4.1 Introduction

Machine learning obtains predictions by constructing a predictive model. Many machine learning algorithms are built on black-box models, as opposed to statistical learning, which is more concerned with the statistical properties of the model, such as the distribution of the variables and its parameters. In this section, we focus on the problem of data- and dimensionality-reduction in Bayesian statistics. Bayesian regression does not assume a fixed optimal solution for a dataset as in the frequentist case, but introduces a distribution over the parameter space. The likelihood function models the information that comes from the data, and the prior distribution models problem-specific prior knowledge. Our goal is to explore and characterize the posterior distribution, which, as a consequence of Bayes' theorem, is a compromise between the observed data situation and the prior knowledge that we assume for the parameters. For very large and high-dimensional datasets and settings where computational resources are scarce, the posterior distribution is hard to obtain. In general, our work focuses on algorithmic approaches that can be implemented in streaming and distributed environments to reduce the underlying problem in order to enhance the scalability of modern Bayesian regression approaches. In this contribution, we particularly concentrate on biomedical applications. Depending on the large-scale high-dimensional problems at hand, our interest centers around a) reducing the number of observations, b) reducing the number of variables, or c) reducing the underlying statistical model. Task a)

comprises approaches such as sketching or coreset methods. We have made various contributions on this topic; see, e.g., Geppert et al. [226] or Munteanu et al. [463]. For more details on this topic, see Section 3.2 in Volume 1 ("Coresets and Sketches for Regression Problems on Data Streams and Distributed Data"). Task b) is particularly relevant for high-dimensional settings in which the number of variables exceeds the number of observations. This is the main problem in many genetic applications. We will present two variable selection approaches for reducing the number of variables, one in the context of a single data source for Bayesian (logic) regression, one in the context of integrating multiple genomic data sources in a Bayesian Cox model. For task c) of reducing statistical models, we transfer the Merge & Reduce principle to maintain statistical summaries in streaming models (Geppert et al. [227]). We can compute the necessary results for a regression model by analyzing them blockwise and combining the summaries of each block in a structured way; more details are given in Section 2.4.3.

2.4.2 Variable Selection

The variable selection topic is one of the central problems in modern statistics. In some fields, researchers are often faced with the problem that the number of variables is larger than the sample size, which is commonly known as the p larger than n problem. In this case, training statistical models directly on the original dataset may lead to many problems, such as overfitting or the inability to use the Ordinary Least Squares (OLS) regression method. Therefore, selecting only those variables that are truly informative becomes a very important step in the process of constructing a model.

A popular approach for variable selection is to make the model sparse by forcing the coefficients of some variables to be zero or converge to zero during the regression process by L_1 or L_2 regularization. Another often-used approach is usually seen in the ensemble algorithm: all variables are first included in the model, and then the variables are ranked according to their importance by a variable importance measure, and the variables that have an impact on the dependent variable are selected after modeling. The following two approaches are well suited for medical applications: a variable importance measure approach employed prior to model building, and a regularized modeling approach using suitable priors and a stochastic search for variable selection.

2.4.2.1 Variable Selection for High-dimensional Binary Data

For high-dimensional binary data, e.g., genetic marker data, interactions between variables are often more important than main effects, which increases the number of variables even further. In this section, we describe how to use so-called *leverage scores* and *cross-leverage scores* as measures of variable importance to select subsets of

explanatory variables while retaining valuable interactions. The leverage and cross-leverage scores are also compared with the more popular variable selection criteria *correlation coefficients* and p-values for univariate linear regressions. In contrast to common variable selection criteria, our approach focuses on variable selection prior to building the model. For more details on our method, see Parry et al. [492].

Obtaining the leverage and the cross-leverage scores requires the calculation of the *hat matrix* of the data matrix, which is a projection matrix that carries the information about the impact of variables instead of observations or responses.
For a data design matrix $\boldsymbol{X}$ and response vector $\boldsymbol{y}$, we set

$$\tilde{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{y}]^T \in \mathbb{R}^{(p+1)\times n} \tag{2.8}$$

and obtain the hat matrix

$$\boldsymbol{H} = \tilde{\boldsymbol{X}}(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}^T \in \mathbb{R}^{(p+1)\times(p+1)}. \tag{2.9}$$

Calculating the hat matrix requires the calculation of the inverse matrix $(\tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}})^{-1}$, which is not a stable calculation (see, e.g., Geppert et al. [226]). An alternative method is to obtain the hat matrix by *QR-decomposition*.

The i^{th} element on the main diagonal of the hat matrix for the first p elements $i = 1, \ldots, p$ is an importance measure of the i^{th} variable, and is also known as its *leverage score* (see, e.g., Drineas et al. [181]). The *cross-leverage scores* are obtained by the off-diagonal elements of a hat matrix (Chatterjee and Hadi [122]) and describe the mutual influence of the i^{th} variable on the j^{th} one for $j = 1, \ldots, p$ and on the response variable for $j = p + 1$.

When selecting the most important variables prior to building the model, one important consideration is how many variables should be included in the pre-selection. Parry et al. [492] point out, based on the available literature, that using $\mathcal{O}(n \ln n)$ is a valid indicator for the optimal number of the pre-selected variables and that choosing that many variables allows the subsampled data matrix to remain at full rank.

For studying the relationship between binary variables as well as their interactions with respect to a response of interest, Ruczinski et al. [550] developed the so-called *logic regression*, an adaptation of a generalized linear regression model. In logic regression, the binary variables are not employed directly as independent variables, but are combined using the logical operators $\wedge$ (and), $\vee$ (or), and the negation !. The resulting Boolean expressions L_i are called logic trees, and the corresponding logic regression model can be written as:

$$g(\mathbb{E}(\boldsymbol{y})) = \beta_0 + \sum_{i=1}^{T} \beta_i L_i, \tag{2.10}$$

where $g(\cdot)$ is the link function of the generalized linear model and L_i for $i \in 1, \ldots, T$ are the Boolean combinations of the binary variables. In our genetic application, we employed the logit link function. The models are fitted with an iterative search algorithm based on Simulated Annealing (see, e.g., [550]).

While logic regression models are well suited for analyzing the association of binary variables and their interactions with a response, a drawback is that they can handle only a limited number of variables. Here, our approach using leverage or cross-leverage scores is helpful, since it reduces the number of independent variables prior to the analysis, while keeping the most influential ones.

We evaluate our approach using simulated data from different scenarios as well as the real HapMap dataset. We compare the leverage and cross-leverage scores for variable selection with correlation coefficients and p-values as selection criteria. In our simulation study, we have created different data scenarios in order to explore the ability of these variable selection measures to select the correct variable when the number of independent variables and the sample sizes vary. We also study these measures for main effects and for the presence of different higher-order interactions.

The genetic HapMap data was collected as part of an international collaboration to develop a haplotype map of the human genome. The subset considered in our work is available in the R-package `SNPassoc` [234]. For our analysis, we considered 7648 human single-nucleotide polymorphisms (SNPs) employed as binary variables for $n = 120$ individuals from two separate ethnic groups. The dependent variable was set to 1 when an individual is from central Europe (CEU), and 0 when it belongs to the Yoruba group (YRI) that inhabits western Africa.

For the simulated data, we show the results by plotting the distributions of leverage scores and cross-leverage scores for the different scenarios (see Parry et al. [492]). The leverage scores perform better in selecting important variables when only main effects are present, whereas cross-leverage scores distinguish variables of higher-order interactions better. A change in the number of variables p did not have much effect on the overall performance of the two measures when the number of irrelevant variables increased. By contrast, when the sample size increases, both scores for the informative variables increase while both measures for the irrelevant variables decrease, meaning that more samples make it easier to find the truly informative variables for both measures. Moreover, we compare the leverage and cross-leverage scores to the correlation coefficients and to p-values.

We also propose and investigate a way of combining cross-leverage scores and leverage scores by selecting the variables using these two metrics separately and taking the union of all selected variables. We show that this combined approach is superior to using just one metric alone (see Parry et al. [492]).

For the HapMap data we present here a raster plot (see Figure 2.21). It is a graphical representation of a matrix containing SNP data, where the rows correspond to individuals and the columns correspond to SNPs. We take subsets of size $\lceil 120\ln(120) \rceil = 575$ of the most important SNPs using cross-leverage scores (CLS), leverage scores (LS), correlations (COR), and p-values, respectively to be comparable. We can easily distinguish the two groups using cross-leverage scores; thus, the selected SNPs can be used to classify individuals of the two groups. Using leverage scores, it is almost impossible to distinguish the two groups. The variables selected using correlation coefficients or

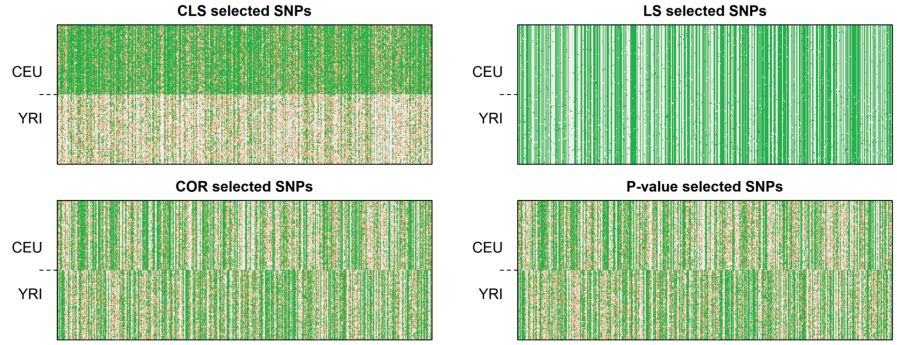


Fig. 2.21: Raster plots of all 120 subjects from two continental regions with respect to a subset of 575 out of 7648 SNPs (green/light pink denotes homozygotic individuals and white denotes heterozygotic individuals), selected using CLS, LS, COR, and p-values, respectively.

p-values show hints of blocks corresponding to the two groups, but much less than those selected by the cross-leverage scores.

Based on our simulation study and the real data example, cross-leverage scores turn out to be a promising tool for variable selection prior to model building, especially in the presence of higher-order interactions, and leverage scores prove useful for selecting main effects. A combination of leverage and cross-leverage scores usually further improves variable selection.

2.4.2.2 Variable Selection for High-Dimensional Survival Data

In the following, we briefly summarize a Bayesian approach that combines the recent progress in the following areas in one model:

1. analyzing time-to-event data in high dimensions
2. variable selection in a high-dimensional setting, and
3. integration of several data sources.

Here, we give a brief overview. For more details on this research, see Treppmann et al. [691].

We usually encounter time-to-event endpoints or survival data in cancer studies. To analyze them, Cox [149] created the semi-parametric proportional hazards regression model that considers the relation between covariates and the hazard function. Therefore, the Cox model has often been applied to low-dimensional data. However, in biological applications with genomic data, we often deal with high-dimensional settings with more variables than subjects. This shows the need for a high-dimensional survival time model. With this in mind, Lee et al. [382] developed a Bayesian version of the Cox model for right-censored survival data, where high dimensions are treated by a

regularization of the regression coefficient vector via Laplace priors. Another contribution to survival prediction for high-dimensional data based on the Cox model in this volume is presented by Rahnenführer et al. in Section 2.5.

A second aspect resulting from a huge number of variables is the need for variable selection. In a high-dimensional setting, common methods like best subset selection as well as backward and forward selection prove to be unsuitable for various reasons. Bayesian techniques provide a good alternative to search stochastically over the entire parameter space, especially as they implicitly address model uncertainty. One example is the stochastic search variable selection (SSVS) by George and McCulloch [224], an approach commonly used in regression analyses. It is a flexible and intuitive procedure that uses data augmentation for the selection task and includes shrinkage.

Moreover, the Bayesian setting provides a way for incorporating additional data sources. The interest in such integrative statistical analyses is growing steadily, as technological progress makes it possible to collect different genome-wide data systematically. The integration of more than one information source can lead to an improvement in the performance of risk prediction models and, therefore, to a more detailed understanding of the biology of diseases. For a recent overview of integrative Bayesian analyses in molecular biology, see Ickstadt et al. [295].

In conclusion, our approach combines the variable selection procedure of George and McCulloch [224] with the Cox proportional hazards model of Lee et al. [382] in one Bayesian model and integrates a further data source by means of an informed prior.

As mentioned, for right-censored survival data in high dimensions with the number of variables p being (much) larger than the number of subjects n, Lee et al. [382] developed a Bayesian variant of the semiparametric proportional hazards model $\lambda(t|x) = h_0(t) \cdot \exp(x^T\beta)$ by Cox [149]. Here, $h_0(t)$ denotes the underlying baseline hazard function, t the survival time of a person with covariable vector $x = (x_1, \ldots, x_p)^T$, and $\beta = (\beta_1, \ldots, \beta_p)^T$ the vector of regression coefficients. By a finite partitioning of the time axis, $0 < s_0 < s_1 < s_2 < \ldots < s_J$ with $s_J > t_r$, $\forall r = 1, \ldots, n$, such that the breaks are points where at least one event occurs, and the last event lies inside the last interval, Lee et al. [382] obtain the following grouped likelihood introduced by Burridge [110]:

$$
\begin{aligned}
L(\mathcal{D}|\beta, h) \propto \prod_{j=1}^{J} \Bigg(& \exp\Bigg(- h_j \cdot \sum_{l \in (\mathcal{R}_j - \mathcal{D}_j)} \exp(x_l^T \beta) \Bigg) \\
& \cdot \prod_{\xi \in \mathcal{D}_j} \left(1 - \exp\left(-h_j \cdot \exp(x_\xi^T \beta) \right) \right) \Bigg) \qquad (2.11) \\
& h_j \sim \Gamma(a_{0j} - a_{0j-1}, c_0) \\
\text{with } a_{0j} = c_0 \cdot H^*(s_j) & \text{ and } c_0 > 0, \quad j = 1, \ldots, J.
\end{aligned}
$$

In this context, $\mathbf{D} = \{(x, \mathcal{R}_j, \mathcal{D}_j) : j = 1, \ldots, J\}$ denotes the observed data, with $\mathcal{R}_j$ being the risk set and $\mathcal{D}_j$ being the event set regarding the j^{th} interval. In case of choosing a

Weibull distribution for the monotonously increasing function $H^*(t)$ (with $H^*(0) = 0$), we obtain $H^*(t) = \eta_0 \cdot t^{\kappa_0}$ with hyperparameters (η_0, κ_0).

For the variable selection, we apply the SSVS procedure of George and McCulloch [224]. Under the assumption that the variances of the regression coefficients of variables included in the model are equal, the prior distribution for β_i conditioned on γ_i has the following form:

$$\beta_i|\gamma_i \sim (1-\gamma_i) \cdot N(0, \tau^2) + \gamma_i \cdot N(0, c_b^2 \cdot \tau^2), \quad i = 1, \ldots, p, \tag{2.12}$$

with small $\tau^2 > 0$ and $c_b^2 > 1$. Following the concept of data augmentation (Tanner and Wong, 1987 [673]), the indicator vector γ states whether the associated variables are included in the model or not.

Inference is based on Markov Chain Monte Carlo (MCMC) algorithms. For updating the full conditional distribution $P(\beta_i|\beta_{-i}, \gamma, h, \mathbf{D})$ with

$$\beta_{-i} = (\beta_1, \ldots, \beta_{i-1}, \beta_{i+1}, \ldots, \beta_p)^T \tag{2.13}$$

we use the special random walk Metropolis-Hastings method with adaptive jumping rules proposed by Lee et al. [382]. Moreover, the conditional distributions $P(\gamma_i^{it} = 1|\beta^{it}, \sigma^{it}, \gamma_{-i}^{it})$ with $\gamma_{-i}^{it} = (\gamma_1^{it}, \ldots, \gamma_{i-1}^{it}, \gamma_{i+1}^{it}, \ldots, \gamma_p^{it})^T$ are derived by means of the Bernoulli distribution. The full conditional distribution $P(h_j|h_{-j}, \beta, \gamma, \mathbf{D})$ with $h_{-j} = (h_1, \ldots h_{i-1}, h_{i+1}, \ldots, h_J)^T$ is approximable by a Gamma distribution. To update β, γ and h iteratively according to the full conditional distributions described above, a Gibbs sampler is appropriate.

In addition to a simulation study, which will not be discussed here, we applied our method to a dataset of glioblastoma multiforme (GBM) patients, retrieved from the Cancer Genome Atlas [471] database. In adults, glioblastoma is the most frequent and the most rapidly growing brain tumor. The used dataset comprises 210 patients and includes survival and gene expression data as well as associated copy number variation (CNV) data, which are used to construct an informative prior. We restrict the analysis to the 1000 genes that show the greatest variance in their values. The underlying assumption is that genes with low variability are probably not well suited to distinguish between patients with a good and patients with a poor survival prognosis. We divide the dataset into a training dataset for the model fitting of 140 patients and a test dataset for the evaluation of 70 patients.

For the analysis, we assume a prior expected number of selected variables of $k = 20$. We construct an informative prior such that the prior inclusion probability π_i^{CNV} of the i^{th} variable is proportional to its standard deviation σ_i^{CNV} of the copy number variation data for the associated genomic region across patients. Thus π_i^{CNV} is defined as

$$\pi_i^{\text{CNV}} = k \cdot \frac{\sigma_i^{\text{CNV}}}{\sum_{j=1}^{p} \sigma_j^{\text{CNV}}}, \quad i = 1, \ldots, p. \tag{2.14}$$

For comparison purposes, we use the non-informative prior $\pi = (k/p, \ldots, k/p)^T$.

In the course of the evaluation, we consider the posterior means and standard deviations of the parameters β and γ. To decide which variables to select, we first determine the mean model size p_m by rounding the average of selected variables per iteration. Then we select the p_m variables with the highest selection probability.

We conduct a combined analysis of five Markov chains, each with a length of 100 000 of which the first 10 % are removed as burn-in.

The simulation result from Table 1 in Section 2 in Treppmann et al. [691] shows that the posterior selection probabilities differ greatly depending on whether the informative or uninformative prior was used. Only three genes are among the $p_m = 10$ (uninformative) or $p_m = 9$ variables (informative prior) with the highest posterior selection probability in both cases.

To evaluate the goodness of the prediction, we consider prediction error curves and determine the integrated Brier score [241, 589] in comparison with the Kaplan-Meier estimator without any covariates (reference approach). This shows that in the case of the informative prior, our model improves the prediction performance relative to the reference approach, while this is not observed for the uninformative prior. The examination of trace plots to the simulated MCMC chains indicates that the chains move quickly into desired regions of the model space and exhibit good mixing performance.

Both in our application to glioblastoma data and in our simulation study, we have shown that the inclusion of a second data source has distinct potential for improvement in terms of prediction quality. However, this is only the case if the second data source provides an informative prior. This requires that variables with an increased prior selection probability tend to be truly associated with the response. However, since this is usually not known in practice, a comparison with the model using an uninformative prior is always appropriate.

Due to the Bayesian modeling, we obtain full inference, especially concerning the posterior selection probabilities. The joint analysis of all variables offers the great advantage that posterior selection probabilities of whole sets of variables can be considered. One example would be a group of genes that has been shown to be particularly influential in previous studies.

Since in MCMC approaches there is usually a trade-off between the computational cost and the accuracy of the results, efficient programming is of particular importance. We recently re-implemented our approach in Python, eliminating some inefficiencies of our previous implementation of the algorithm in R.

2.4.3 Merge & Reduce for Statistical Models

Datasets with a massive number of observations have become more and more common, making scalability a major challenge for modern data analysis. For many statistical methods, these amounts of data lead to an enormous consumption of resources. A prominent example is linear regression, an important statistical tool in both Bayesian

and frequentist settings. On very large datasets, regression analysis becomes increasingly demanding with regard to running time and memory consumption, making the analysis tedious or even impossible. We propose a method called Merge & Reduce to address these scalability limitations in regression analyses. Merge & Reduce is well known in computer science and has mainly been used for transforming static data structures to dynamic ones with little overhead (Bentley and Saxe [56]). Instead of reducing the data to approximate the full dataset with respect to some model, we propose using the statistical models derived from small batches as concise summaries. Combining these statistical models via the Merge & Reduce framework, we can turn an offline algorithm into a data stream algorithm.

Here, we focus on streaming to deal with massive datasets, where a data stream algorithm is given an input stream of items, like numerical values, vectors, or edges of a graph at a high rate. The algorithm is allowed to make only one single pass over the data. As the items arrive one by one, it maintains a summary of the data that was observed so far in the form of, say, subsample or a summary statistic. Despite our focus on a streaming-setting, we stress that the Merge & Reduce scheme can also be implemented in distributed environments.

Our contribution is to develop the first Merge & Reduce scheme that works directly on statistical models. We show how to design and implement this general scheme for the special cases of (Bayesian) linear models, Gaussian mixture models, and generalized linear regression models in Geppert et al. [227]. Here, we will restrict ourselves to Bayesian linear models and evaluate the resulting streaming algorithm on simulated datasets. We demonstrate that we obtain stable regression models from large data streams and that the Merge & Reduce schemes produce little overhead.

2.4.3.1 Method

In our Merge & Reduce method for statistical data analysis, we iteratively load as many observations into the memory as we can afford. On each of these blocks, we apply a classical algorithm to obtain, say, the parameters of a statistical model, some (sufficient) statistics or a summary of the presented data; in short, a model. Models are merged according to certain rules, eventually resulting in a final model that combines the information from all subsets. Merge & Reduce leads to stable results, where every observation enters the final model with equal weight, thus ensuring that the order of the data blocks does not bias the outcome toward single observations.

In order to design a streaming algorithm for a specific statistical analysis task, we need to choose an appropriate model as a summary statistic for each block of data. The two main ingredients that we need to implement for this particular choice of a model are called *merge* and *reduce*.

1. Let M_1, M_2 be the models obtained from the analysis of data blocks B_1, B_2, then the output of **merge**(M_1, M_2) is a model M for the union $B_1 \cup B_2$ of the input data blocks.

2. Let M be a model for data block B that has become too large, i.e., $|M| \geq 2T$ for some threshold T (e.g., by repeated merge operations), then **reduce**(M) computes a model $M^{'}$ of size $|M^{'}| \leq T$ for the same block B.

Summarizing the statistical analysis and implementing the Merge & Reduce functions on statistical models are not trivial undertakings. The approach heavily depends on the statistical method employed and on the representation chosen to store the model. Here, we present the novel, general concept and discuss how to design the Merge & Reduce functions for the example of linear regressionindexRegression!linear in the Bayesian setting.

We first describe how the Merge & Reduce functions interact in a structured way to perform the statistical analysis task on the data block-by-block while maintaining a model for the whole subset of data presented so far. The data structure consists of $L = O(\log(n/n_b)) = O(\log n)$ buckets for a sufficiently small block size n_b to fit into the main memory of the machine. The buckets store one statistical model each. Initially, they are all empty. One bucket, the working bucket B_0, is dedicated to store the model for the current batch of data, while each of the other buckets B_i stores one model on its corresponding level $i \in \{1, \ldots, L\}$ of a binary tree structure formed by the merge operations, see Figure 2.22. The data structure works in the following way. First, we read one block of data of size n_b. We perform the statistical data analysis on this block only. The model that summarizes the analysis is stored into B_0. We begin to propagate the model in the tree structure from bottom to top by repeatedly executing Merge & Reduce operations on each level. If B_1 is empty, then we just copy the model from B_0 to B_1 and empty B_0. Otherwise, we have two models that are siblings in the tree, so we merge the two into B_0, empty B_1 and proceed with B_2. Again, if it is empty, the model from B_0 is stored in B_2 and the propagation terminates. Otherwise, we have two siblings that can be merged and propagated to the next higher level in the tree. In general, the propagation stops as soon as the bucket on the current level is empty. When this happens, the update of the data structure has completed, and we can move on to reading and analyzing the next block of input data. This is repeated until the end of the stream. Notice that except for the additional working bucket, we need to store at most one bucket on each level at a time, since two siblings are merged immediately.

A linear regression model is given by

$$Y = X\beta + \epsilon, \tag{2.15}$$

where $Y \in \mathbb{R}^n$ is the dependent variable and $X \in \mathbb{R}^{n \times d}$ is the design matrix containing the observations $x_1, \ldots, x_d$ of the independent variables. The error term ϵ is assumed to be unobservable and is usually modeled by a normal $N(0, \sigma_\epsilon)$ distribution, $\beta \in \mathbb{R}^d$ is the unknown parameter vector of regression coefficients that we wish to estimate. In Bayesian regression, β is assumed to be random and follows a distribution. Interest

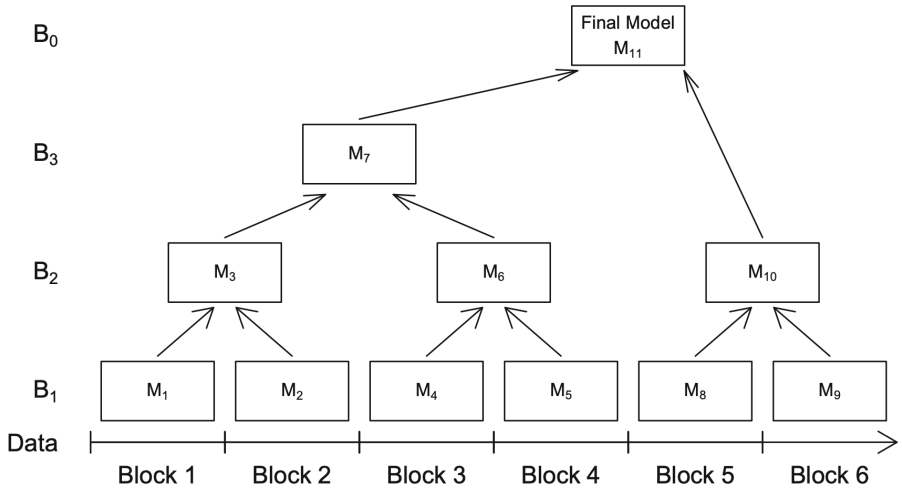


Fig. 2.22: Illustration of the Merge & Reduce principle taken from Geppert et al. [227]. The data is presented in the form of a stream and subdivided into Blocks 1 through 6 of equal size. Models M_1 to M_{11} are numbered in order of their creation throughout the execution. Arrows between models indicate the Merge & Reduce operations. Sibling models are deleted right after their parents' creation. Thus, only one model is stored on each level, i.e., in buckets B_1 to B_3, at a time. The working bucket B_0 acts on all levels, eventually holding the final model after postprocessing at the end of the stream.

centers around the posterior distribution of the parameters that can be written as

$$p_{post}(\beta|X, Y) \propto \mathcal{L}(Y|X, \beta) \cdot p_{pre}(\beta). \tag{2.16}$$

Thus the posterior distribution p_{post} can be seen as the product of the prior distribution p_{pre} and the likelihood function $\mathcal{L}$ of the parameters. In many cases, the posterior distribution cannot be obtained directly by analytical means, but must be approximated by some sampling approaches. The most popular method is to apply Markov Chain Monte Carlo (MCMC) random sampling, but this requires a very large number of simulations. When the sample size is also very large, such sampling simulations are demanding in terms of computer memory and computing speed.

We will apply the Merge & Reduce method to Bayesian linear regression. Since we use the MCMC method to approximate the posterior distributions of the parameters β, for each estimated parameter β_j, we collect the mean $\bar{x}_j$, the median $\tilde{x}_{.5}$, the lower and upper quartiles $\tilde{x}_{.25}$ and $\tilde{x}_{.75}$, 2.5 % and 97.5 % quantiles $\tilde{x}_{.025}$ and $\tilde{x}_{.975}$, and the standard deviation σ_j of the posterior distributions. Then, the collected statistics can be summarized as

$$S = (\bar{x}_1, \ldots, \bar{x}_d, \tilde{x}_{p,1}, \ldots, \tilde{x}_{p,d}, \sigma_1, \ldots, \sigma_d), \tag{2.17}$$

where $p \in \{0.025, 0.25, 0.5, 0.75, 0.975\}$. We use a weighted average of the statistics to merge these statistics vectors, and the weights are dependent on the size of the sample.

2.4.3.2 Simulation and Results

The main parameters used to generate datasets for the simulation study are the number of observations n, the number of variables d and the standard deviation σ_ϵ of the error term ϵ. Different numbers of observations per block n_b are also chosen. The setup of the simulation study and the parameter values are similar to those in the simulation study in Geppert et al. [226].

In this study, all assumptions of a linear regression model are met. A varying fraction of the variables has an influence (large or small) on the dependent variable, while the remainder is not important for the explanation of Y.

We evaluate the Merge & Reduce approach by calculating the squared Euclidean distances e_m^2 of the statistics in S, specified in Equation 2.17, between the original model and the Merge & Reduce model for all simulation models $m = 1, \ldots, M$. If the Euclidean distance is close to 0, then the Merge & Reduce approach approximates the results of the original model accurately. We see in Figure 2.23 that, as the ratio $\frac{n_d}{d}$ increases, the difference between the medians, obtained from the Merge & Reduce approach and from the Bayesian original model, evaluated by their squared Euclidean distance, is quite small. The majority of the squared distances is close to 0. For more details on the results of other summary statistics of the posterior distributions, see Geppert et al. [227].

2.4.3.3 Conclusion and Outlook

Merge & Reduce is suitable for Bayesian regression models. The goodness of the approximation depends on the ratio of observations per block and variables $\frac{n_d}{d}$ and the goodness of fit of the original model. The first condition can easily be controlled by the data analyst, especially in a setting with large n. The second condition may require care when building the model.

For the implementation of the Merge & Reduce approach in principle, it is only necessary to choose an appropriate statistical model and to implement Merge & Reduce operations for this specific type of model. However, the design of such operations is not trivial in general. In particular, we showed in Geppert et al. [227] how to design such operations for the case of Bayesian linear regression, Gaussian mixture models, and generalized linear models. Implementing the Merge & Reduce approach for the Bayesian Cox model is a next step in future work.

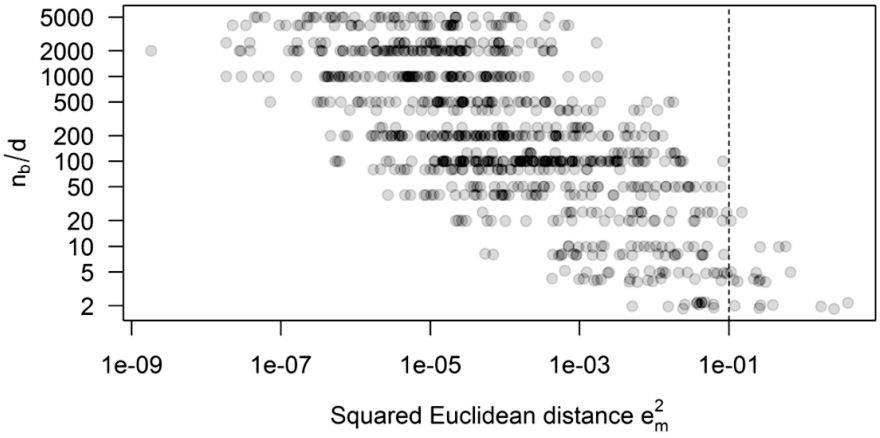


Fig. 2.23: The figure, taken from Geppert et al. [227], shows a scatterplot of the effect of observations per block per variable $\frac{n_b}{d}$ on squared Euclidean distances e_m^2, $(m = 1, \ldots, M)$ between posterior medians for the Bayesian Merge & Reduce approach. The *x*- and *y*-axes are drawn on a logarithmic scale, and observations are drawn as partially transparent points: gray points mean single observations; black points represent multiple observations at roughly the same location. The vertical dashed line is at 0.1.

2.4.4 Overall Conclusion

We are concerned with several issues that arise when dealing with insufficient computational resources. In this contribution, we show ways of reducing high-dimensional data and simplifying complex models in the context of biomedical applications.

In the first two sections, we introduce variable selection strategies in two different high-dimensional datasettings. In the first scenario, we describe a variable importance measure approach for genetic SNP data, employed prior to model building. In the second scenario, we formulate a variable selection strategy for high-dimensional time-to-event data integrating several data sources in the context of a Bayesian Cox model. In the third section, we use the Merge & Reduce approach for massive data to analyze and build statistical models on small batches separately and recombine them in order to cope with the problems of limited running time and insufficient memory.

All approaches in this contribution improve the efficiency of statistical learning when facing the limited computational resources of host devices. They reduce the computation time, energyconsumption, and memoryusage of learning individual models without losing the high accuracy of the model. This, in turn, allows us to analyze more complex models on massive data with an efficient use of computational resources.

2.5 Survival Prediction and Model Selection

Jörg Rahnenführer
Michel Lang
Jakob Richter

Abstract: Survival analysis comprises statistical methods for time-to-event data. The main prediction tasks include the estimation of the influence of prognostic factors for, say, medical treatments, and the modelling and prediction of survival times using regression models. In recent years, in molecular medicine, many omics technologies have been developed, generating complex high-dimensional genetic data that can be used as predictors.

For such complex tasks, the selection of the best prediction method out of a large set of candidates, along with potential feature selection and hyperparameter optimization, represents an optimization task under resource constraints. In this section, approaches for tackling the model selection problem in survival analysis are presented, specifically using Bayesian optimization and addressing feature selection for high-dimensional data.

2.5.1 Introduction

In medicine, times to events are compared between groups to estimate the effect of prognostic factors and medical treatments, and regression models are used to model and predict survival times of cells, animals, or patients. For two decades, high-dimensional genetic and genomic variables have been generated and analyzed as potential predictive and prognostic factors in biological and medical scenarios. The very large number of variables requires developing and using tailored methods to describe the complex relationships. Popular modeling approaches are based on penalized regression methods, gradient boosting methods, survival trees, and survival forests, often combined with suitable feature selection methods.

In recent years, machine learning approaches were used to find the best survival method from a large set of candidates. Efficient approaches are required, since it is crucial that runtimes especially in resampling scenarios with many repeated estimation tasks be kept short, especially for complex high-dimensional predictor settings. In CRC 876, we applied modern Bayesian optimization (BO) [303] techniques to efficiently identify the best survival prediction method, by modeling the relationship between the choice of the survival prediction method (as well as its hyperparameters) and its performance or quality, using so-called surrogate functions. On several lung cancer datasets the new approach was superior to established benchmark approaches [367,

369]. After a short introduction into the analysis of time-to-event data in Section 2.5.2, model selection for survival analysis is discussed in Section 2.5.3. To solve this task, various R packages were implemented both for the general candidate selection and for parallelization, as presented in Section 2.5.6.

The same principle idea was used in a scenario, where survival predictions for a specific cancer dataset are to be improved, by adding data from similar datasets. This is a frequent situation in, say, cancer survival analysis, where patient numbers in clinical trials are limited due to ethical, financial, and administrative reasons, but similar treatments are applied, e.g., in other clinical centers. However, simply adding similar datasets to the one of interest potentially deteriorates the predictions, due to structural differences between the datasets. Instead, one can estimate dataset-specific weights that determine how strong these datasets should be considered. In CRC 876, we applied BO to determine such optimal weights for inclusion of the respective observations in appropriate weighted likelihood-based modelling approaches [531], as shown in Section 2.5.4 below. In two other projects related to feature selection, we developed improved methods based on two-fold subsampling schemes [383] and benchmarked filter methods against each other for high-dimensional data [95]. These analyses are described in Section 2.5.5.

2.5.2 Analysis of Time-to-Event Data

Survival analysis, also called event-time analysis, deals with the analysis of times to certain events and is used in many application fields. In medicine, the overall survival (OS) of patients is often of direct interest. Alternatively, Progression-Free Survival (PFS) is frequently analyzed, which includes Event-Free Survival (EFS) and recurrence-free survival. An important property of survival data is that they are often not fully observable, such as when patients in a clinical trial have not yet experienced the event of interest at the time the trial ends and the data is analyzed. This situation is called right-censoring, since for patients without an observed event, the survival time must be greater than or equal to the time until the end of the study. Depending on the type of missing information, many other censoring mechanisms are defined and considered in the analysis techniques.

Specialized statistical methods for analyzing survival data have been developed and are widely used in literature and in practice. Most prominent are the Kaplan-Meier estimator for estimating survival curves under right-censoring, the log-rank test for comparing survival between patient groups, and the Cox proportional hazards model [149] for estimating survival dependent on a number of explanatory variables, such as tumor size or age in oncological studies.

In regard to the evaluation of performance, seemingly obvious approaches lead to wrong interpretations of the results. For example, simply predicting the event indicator that indicates if a patient has survived until the end of the study, neglects the different

time intervals that have passed since the patients entered the study. By contrast, methods based on hazard rates that model the instantaneous failure rates at different time points can cope with the censoring mechanisms. Alternatively, parametric likelihood methods that consider the missing information can also be used.

For evaluating the prediction accuracy of survival models, several suitable measures have been developed. Concordance statistics, in particular Harrell's C-index and the area under the (time-dependent) ROC curve, are the most popular measures. However, they consider only the discrimination ability of a survival model and not the calibration. This means that monotone transformations of predicted values of survival outcomes do not change the concordance score, which limits the interpretability of the score for clinicians. Alternatively, the Brier score is also widely used. It considers both calibration and discrimination, but interpretation is also difficult. An advantage is that it can be related to a time-specified horizon. A discussion of these important properties and an adaptation of the Brier score can be found in Kattan and Gerds [313].

In preclinical and clinical studies, genetic factors are of interest, and modern high-throughput technologies provide many thousand potential explanatory variables. Even the popular but controversial rule of thumb that the number of events per variable should be at least 10 cannot be used as a basis for sample size planning. Instead, tailored statistical and machine learning approaches are required. Aspects to consider for model selection in this scenario are discussed in the next subsection.

2.5.3 Model Selection for Survival Analysis

Model selection in survival analysis, compared with model selection in classical machine learning setups such as regression or classification, presents numerous additional challenges.

First, instead of having to solve a learning task with many observations (e.g. patients) and comparatively few variables, in survival analysis we often face a low sample size problem. Even worse, with the rise of omics technologies, thousands to hundreds of thousands of genetic features need to be included in the analysis to be able to identify the most important genes. However, most machine learning or statistical learning algorithms have been designed and heavily optimized for a large sample size n, and usually have worse than quadratic runtime in the number of features p. For this reason alone, we often face runtime issues in the $n \ll p$ scenario.

Second, a dual objective is often pursued, and the predictive performance of the models is not the only target criterion. Instead, it is desired to identify the important features (clinical covariates, genetic dispositions, or genes) in the given medical context. A good predictive performance often ensures that the model describes the data in a meaningful way, which is the prerequisite for extracting a set of important features. This restricts the analysis to models that either come with an embedded feature selection or models that still work reasonably well after a feature filter has been applied.

Third, all performance measures in survival analysis require a large enough test set to yield performance values that lead to reliable statements. For example, the popular C-index mentioned above assesses the ranking of the predictions for survival in the test set by comparing it with the true, observed ranking of survival times (while correcting for censoring). Obviously, having too few observations in the set results in a high variance of the performance estimator. Since usually only a few hundred observations are available in total per dataset, the number of repetitions must be increased during resampling in order to account for the larger variance. Of course, this exacerbates the runtime problems one is already facing.

These points taken together form a hard tuning problem with the following characteristics:

1. The models form a black box from the perspective of the tuner, as there are no known derivations. Therefore, the optimization problem itself is also called "black box".
2. To assess the predictive performance of a hyperparameter configuration θ and its resulting model, the data needs to be split into a training set and an independent test set. This introduces stochastic components into our tuning problem at the latest (some learners are non-deterministic either way).
3. The search space spanned by the hyperparameters to be tuned usually includes both numerical and categorical variables. This precludes the use of many tuners derived from discrete and steady optimization.
4. Each model fit is potentially resource demanding, in terms of computational time, memory requirements, or communication costs. The key word here is "potentially". Some models, e.g., a simple Cox model augmented with an aggressive feature filter, can easily be fitted in less than a minute even with $n = 200$ observations and $p = 10^6$ variables (features). Other learners, such as support vector machines, require a complete day for the same task on the same hardware while simultaneously consuming several orders of magnitude more memory. Obviously, the resource requirements are very heterogeneous, which should be taken into account during the mandatory parallelization.
5. Last but not least, during hyperparameter optimization, we generally have to deal with an additional type of censoring (besides the censoring of the survival times): It is not unusual that the learner implementations crash from time to time due, say, to numerical problems. And since the tuning is usually distributed on larger computation sites with shared and contested resources, computational jobs can hit a wall time and be killed by a scheduler. In such a case, the missing performance score must be imputed with a number to continue with the tuning, and it is unclear which value to choose.

Over the last decade, special strategies addressing the difficulties of hyperparameter optimization have emerged. An overview is given by Bischl et al. [68]. Roughly speaking, hyperparameter optimization is about finding the configuration θ of a model, which

leads to the best predictive performance (evaluated on an independent test set). If the evaluation of a single configuration is sufficiently expensive with regard to computational resources like runtime, every evaluation counts, which also means that rather wasteful optimization methods are not applicable. This applies, among others, to Evolutionary Algorithms (EAs). EAs usually require many hundreds of configurations before being able to make the first targeted decisions. Instead, a tuner that optimizes more aggressively from the start is needed.

One tuner that addresses all the problems of the outlined expensive black-box optimization problem is *iterated F-racing* [421]. The basic idea of F-racing is to race a population of configurations against each other, and to eliminate in each iteration candidates that are underperforming based on a Friedman test. Iterated F-racing extends this approach by assuming a probability distribution over the search space. This distribution gets updated iteratively so as to be centered around some elite configurations. We applied this tuning approach to a broad range of survival pipelines (consisting of the feature filter and the survival model) [369]. The benchmark considers 12 different datasets of four breast cancer cohorts where each dataset consists of clinical and/or genetic variables (features). The architecture of the pipeline, i.e. the choice of filter and the choice of model, is encoded as virtual hyperparameters passed to the tuner. This way, dominated combinations of filters and models are getting fewer evaluations, giving the tuner more opportunity to exploit hyperparameters of more promising combinations. As a baseline, four reasonable approaches that are popular in practice but are arguably less computationally intensive have been evaluated. To the best of our knowledge, this was the largest benchmark of survival models up to that point. In comparison with the baseline approach, the tuning yields significantly better results in terms of the C-index. The caveat is the effort to archive the results: with more than 10 000 hyperparameter evaluations, the tuning cannot be applied easily on new data or cohorts.

Another tuner which perfectly fits the requirements of hyperparameter optimization in survival analysis is Model-Based Optimization (MBO). Its performance has been verified by Lang [367] where the benchmark study from Lang, Kotthaus, Marwedel, Weihs, Rahnenführer, and Bischl [369] has been extended, with more datasets, more filters, more models, and more time budget.

Figure 2.24 visualizes the survival probability in the included cohorts. Although the studies are all on lung cancer, and share the same set of clinical and genetic features, they differ considerably with respect to survival times. This is a frequently observed characteristic, and makes the careless merging of the datasets into a larger dataset with more observations inappropriate. As a result, in this domain, it is usually not possible to configure a single model to perform sufficiently well on all cohorts. Instead, for each cohort, tuning starts from zero. One goal of the analysis was to thin out the portfolio of methods to consider for a new tuning run. If, e.g., only two pipelines consisting of a filter and a model have to be tried, the computational effort required for tuning is significantly reduced, rendering the tuning for new cohorts on a single computer

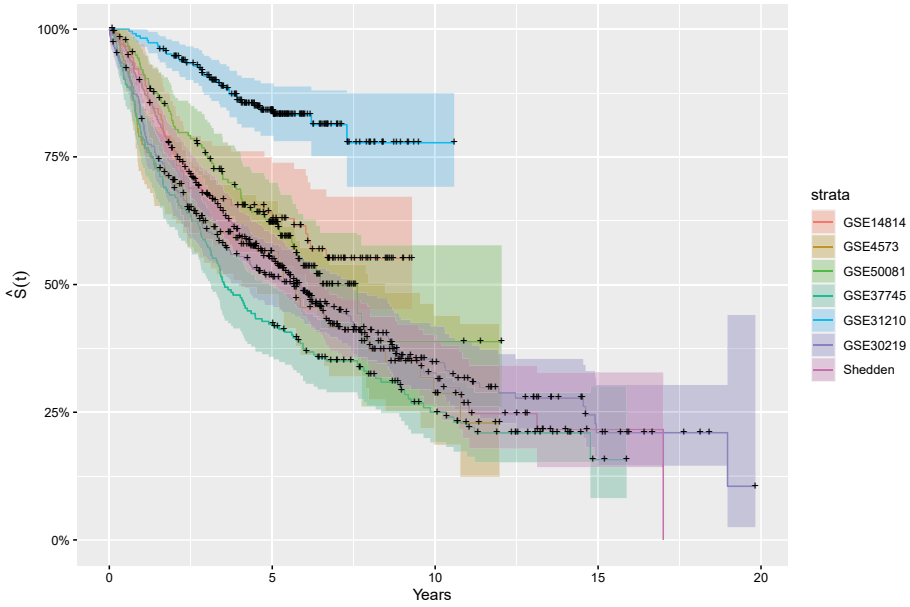


Fig. 2.24: Plot of Kaplan-Meier estimators including confidence bounds for the survival time $S(t)$, stratified by the cohorts that are included in [367]. In the Kaplan-Meier plots, the time t is plotted against the estimated proportion of patients still alive at t. Lines represent survival curves of the seven cohorts. A vertical drop in a curve indicates an event, and a plus on a curve means that an observation was censored at this time.

possible and therefore applicable for practice. This has been systematically analyzed by Lang [367]. Parts of the results are summarized in Figure 2.25. Additionally, the mean ranks of filters and learners have been analyzed and revealed the following important take-home messages in the context of the datasets analyzed:

- If one base learner has to be chosen, random survival forests perform best on average.
- One of the most popular approaches due to its embedded feature selection–fitting a Cox proportional hazards model with a LASSO penalty (L_1)–performs the worst on average.
- Tuning over multiple base algorithms jointly with MBO results in the best performance on average.
- Tuning each pipeline individually and picking the best performing pipeline (approach BenchOpt in Figure 2.25) in a second step is not only a waste of computational resources; it also leads to overoptimistic performance estimates. As each tuning run is stochastic, and the pipelines often perform comparably well, picking the best configuration is determined by the stochastic noise to some degree. This

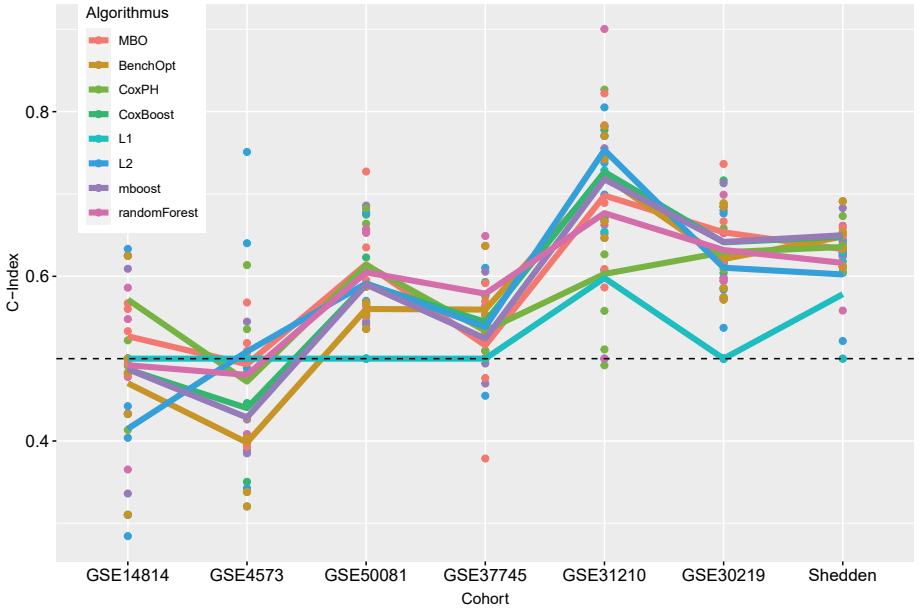


Fig. 2.25: Resulting average C-index on independent test sets of multiple algorithms, stratified by cohort. All base algorithms are individually tuned together, jointly with the choice of filter and the filter's hyperparameters. MBO tunes over all filters and algorithms simultaneously. BenchOpt expresses the resulting performance on an independent test set after picking the best performing base algorithm on the training data [367].

is in particular a very alarming result, as the described manual benchmarking is common practice.

The heterogeneous runtimes (or more general, the heterogeneous resource demands) have been addressed by Richter, Kotthaus, Bischl, Marwedel, Rahnenführer, and Lang [530] and Kotthaus et al. [346]. Instead of fitting only a single surrogate model, guiding the optimization to areas with the best predictive performance, multiple surrogate models are fitted in each iteration. On the one hand, the usual surrogate based on the observed predicted performance is calculated. On the other, one or more surrogate models for computational resources are fitted, e.g., one surrogate for the runtime and one surrogate describing the memory consumption. As a result, we can query the models for the estimated predictive performance and the estimated resource demands for all hyperparameter configurations. All the information is fed into a scheduler that selects a subset of the configurations and maps them to multiple CPUs or workers based on their priority (as derived from the estimated predicted performance) while minimizing the idle times (based on the estimated runtime).

2.5.4 Weighted Subgroup Selection for Survival Analysis

Obtaining a reliable prediction model for a specific cancer subgroup or cohort $s^\star$ is often difficult due to a limited sample size and, in survival analysis, due to potentially high censoring rates. Sometimes similar data from other patient subgroups is available, e.g., from other clinical centers. Simple pooling of all subgroups can decrease the variance of the predicted parameters of the prediction models, but also increase the bias due to heterogeneity between the cohorts.

Different approaches exist to improve the predictive quality by including data from other patient subgroups in a weighted fashion. One possible way is to include one further weighted subgroup, as proposed by Weyer and Binder [726]. Alternatively, individual weights for each patient can be estimated from the training data, as described by Bickel et al. [63]. The idea is that weights match the joint distribution of the combined data to the distribution in each subgroup, such that a patient who is likely to belong to the target subgroup receives a higher weight in the subgroup-specific model. Weights correspond to the conditional probability of belonging to the target subgroup $s^\star$ given the observed covariates and outcome divided by the prior probability for $s^\star$. The former is estimated from the training data by multi-class classification, and the latter by the relative frequency of $s^\star$.

The goal is to optimize the predictive performance of our model for the target subgroup $s^\star$. Including data from additional subgroups in the training data should increase the predictive performance of the target subgroup. Accordingly, this forms a combinatorial optimization problem where additional subgroups must be chosen to maximize the predictive performance.

However, completely abstaining from using certain subgroup data seems overly drastic since there might be relevant information contained in each additional subgroup data. Luckily, most machine learning methods and also those that can be used for time-to-event data allow observational weights. This allows us to give a low weight to observations that do not represent our problem. However, finding an optimal weight for each observation is exceedingly complex. Instead, we introduce subgroup weights as presented by Richter et al. [531]. The observation weight is then determined by the subgroup membership of each observation. This enables the inclusion of certain subgroups with a specific weight. Hence, including additional data in a weighted way might lead to a better solution than the binary choice of including a subgroup with full weight or not at all.

By introducing subgroup weights, we relaxed the combinatorial problem into a numerical optimization problem. However, setting those subgroup weights in an optimal way remains a difficult optimization problem. First, each additional subgroup leads to a further weight parameter that has to be chosen optimally. Second, the evaluation of a weight parameter combination can take fairly long, since the datasets themselves tend to be rather large, especially when they include high-dimensional genetic mea-

surements in the scenario of survival analysis. In this case, it becomes infeasible to try out many weight parameter combinations in order to find an optimal one.

Therefore, we can apply state-of-the art optimization methods for expensive black-box problems such as MBO (model-based optimization) in order to find the optimal subgroup weights without the cost of having to evaluate hundreds of different weight parameter combinations. For our evaluation, we optimize the subgroup-specific weights $w^{(g)}$ in the weighted Cox model. Note that in Section 2.5.3 a study was reported where random survival forests performed on average better than fitting a Cox proportional hazards model with a LASSO penalty. However, here we use the much more frequently used penalized regression approach to evaluate the potential improvement due to the weighted analysis.

Weighted Cox Model Assume the observed data of the patient i consists of the tuples (t_i, δ_i), the covariate vectors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^{'} \in \mathbb{R}^p$, and the subgroup membership $s_i \in \{1, \ldots, S\}$ with S being the total number of available subgroups, and $i = 1, \ldots, n$. t_i denoting the observed time of patient i, the minimum of the event time, and the censoring time. δ_i indicates whether a patient experienced an event ($\delta_i = 1$) or was (right-)censored ($\delta_i = 0$). As mentioned above, one of the most popular regression models in survival analysis is the Cox proportional hazards model [149]. It models the hazard rate $h(t|\boldsymbol{x}_i)$ of a patient at time t as

$$h(t|\boldsymbol{x}_i) = h_0(t) \cdot \exp(\boldsymbol{\beta}^{'} \boldsymbol{x}_i) = h_0(t) \cdot \exp\left(\sum_{j=1}^{p} \beta_j x_{ij}\right),$$

where $h_0(t)$ is the baseline hazard rate, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^{'}$ is the unknown parameter vector. The parameters are estimated by maximizing the partial log-likelihood [326, Chapter 8.3]. A version of the partial log-likelihood that uses observation weights is introduced in [726]:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \delta_i w_i \left(\boldsymbol{\beta}^{'} \boldsymbol{x}_i - \ln \left[\sum_{k=1}^{n} \mathbf{1}_{(t_i \leq t_k)} w_k \exp\left(\boldsymbol{\beta}^{'} \boldsymbol{x}_k\right) \right] \right). \tag{2.18}$$

Instead of an individual weight for each patient, we introduce an individual weight for each subgroup. Therefore, we assign the same weight to each patient of the same subgroup:

$$w_i = \begin{cases} 1, & \text{if } s_i = s^{\star} \\ w^{(g)}, & \text{if } s_i = g,\ g \in \{1, \ldots, S\} \setminus s^{\star} \end{cases} \tag{2.19}$$

where $w^{(g)} \in [0, 1]$ is the specific weight for the subgroup g, and $s^{\star}$ is the subgroup for which we want to obtain predictions. Patients for subgroup $s^{\star}$ enter with full weight 1 in the prediction model.

Standard subgroup analysis is based only on the patients in the subgroup of interest (target subgroup $s^\star$), which corresponds to $w_i = 0$ for all patients not belonging to $s^\star$. A combined model that pools patients from all subgroups corresponds to $w_i = 1$ for all patients.

In high-dimensional settings where the number of covariates p is typically much larger than the sample size n, standard maximum likelihood cannot be used for parameter estimation, since it does not result in a unique solution. Therefore, we add a LASSO penalty [677] to the partial log-likelihood. Lasso regression performs feature selection and yields a sparse model solution. The resulting maximization problem of the penalized partial log-likelihood is given by

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left\{ l(\boldsymbol{\beta}) - \lambda \cdot \sum_{j=1}^{p} |\beta_j| \right\}.$$

The LASSO penalization parameter λ is optimized through an internal 10-fold cross-validation.

Evaluation We are interested in maximizing the predictive performance for a target subgroup $s^\star$. The predictive performance of the weighted Cox model is evaluated using the C-index. For the evaluation of the model for a given target subgroup $s^\star$, a dataset that contains S subgroups, and a subgroup weight vector $w = (w^{(1)}, \ldots, w^{(S-1)})$, we conduct a modified 10-fold cross-validation. The validation data should only contain the target subgroup, because we are only interested in the predictive performance on the target subgroup. In order to obtain the 10 necessary splits for the cross-validation, we only divide the data of the target subgroup into 10 chunks. To obtain the prediction for one chunk, all remaining 9 chunks plus all observations from the additional subgroups are combined to the training dataset. By performing the modified cross-validation, we obtain an estimation on the C-index for the given combination of dataset, target subgroup and subgroup weight vector.

Now, the goal is to find the subgroup weight vector that maximizes the C-index. This optimization problem can be solved with MBO, with a search space $[0, 1]^{S-1}$ that directly maps to the weight vector. The acquisition function that selects the next weight vector to be evaluated should take into consideration that results are not deterministic. Therefore, we proposed the augmented expected improvement [288], which is well suited for such scenarios. For the Gaussian process regression within the Bayesian optimization, we proposed the Matern 3/2 kernel with an estimated *nugget effect* to account for the noisy response of our objective.

The benefit of optimizing the subgroup weights is twofold: First, the resulting optimal subgroup weight vector does not only maximize the C-index for the target subgroup; it also allows drawing conclusions about the similarity of certain subgroups. If a certain subgroup weight is small, it can be assumed that this subgroup does not have a similar relationship between the explanatory variables and the event times as

the target subgroup. Second, as shown by Richter et al. [531], the predictive performance of the method does not deteriorate if additional subgroups are included that contain inconsistent data.

Benefits could arise from using the penalization of the weights, which would allow researchers to completely exclude data with weights close to zero. Then the model becomes computationally cheaper and possibly more stable. Finally, the presented approach can be used for any situation where data is pooled from different cohorts and a machine learning method is used that supports observational weights.

2.5.5 Feature Selection for High Dimensional Data

The problem of feature selection is particularly important in the domain of high dimensional data, as already described in detail in Section 2.5.3. One challenging problem in this context is the stability of feature selection. Some learners can be restricted to using only a small subset of the thousands of available features, and learners can be combined with a feature filter to achieve the same in a generic fashion. However, the set of selected features is often highly variable. For example, if a Cox proportional hazard model is extended with an L_1 penalty λ tuned to include only up to 20 features in a 3-fold cross-validation, the resulting three sets of selected features can be pairwise disjoint. This has a simple reason: if two features x_1 and x_2 are highly correlated, they are also comparably good predictors. If the model has to choose between x_1 and x_2, a few observations can tip the scales in one direction or the other. If the dataset is now resampled and these observations are removed from the training set, the scale can easily swing in the other direction. This is particularly annoying because in this way no features can be reliably selected for a later analysis, such as a biological analysis.

Lee et al. [383] tackle this problem in two ways: First, a special extension to the LASSO regression is used. The preconditioned LASSO [495] is a two-step procedure designed to address the problems of high bias in LASSO estimates. Second, the preconditioned LASSO is embedded in a two-fold subsampling procedure to improve the stability of the feature selection via model averaging and extra shrinking of covariates based on the selection probability in the inner subsampling.

The approach has been applied to datasets on neuroblastoma, lung adenocarcinoma, and breast cancer. Both predictive performance (measured by the C-index) and stability (measured by the Jaccard index and the Kuncheva index) are improved. However, the comparison with popular univariate selection methods does not provide a clear picture.

Another take on this topic was presented by Bommert et al. [95], where more than 20 filter methods are benchmarked against each other for high-dimensional data. Although this work is based on classification data, there is no reason why the core results should not be transferable to survival problems, and confirming this is currently work in progress. One key result is shown in Figure 2.26. There are clear groups of

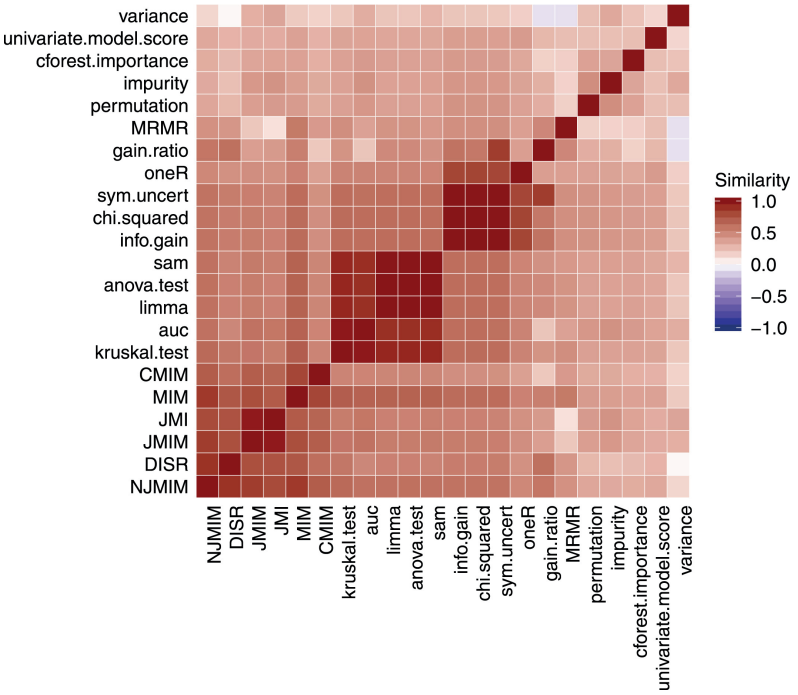


Fig. 2.26: Rank correlations between the feature selection order, for all pairs of a large set of filter methods, averaged across several datasets by the arithmetic mean. The filter methods are ordered by average linkage hierarchical clustering using the mean rank correlation as a similarity measure [95].

feature filters available. Filters from the same group are expected to give very similar results across different datasets. Therefore, instead of including more than 20 filters into the machine learning pipeline, it is sufficient to thin out this portfolio to a smaller set. Additionally, the filters have been analyzed regarding performance and stability to provide general recommendations for feature filtering in high-dimensional settings.

2.5.6 Software

Many machine learning frameworks exist that can be conveniently employed for model selection or feature selection. However, most of these frameworks have a strong focus on classification and regression. Support for survival analysis is often not existent or insufficient. For proper evaluation, two frameworks have been extended to support time-to-event data.

First, the R package `mlr` [69] has been extended with an object for survival tasks, including the most common survival learners and survival measures. By building upon

the existing infrastructure for resampling and tuning, survival learners can be tuned with state-of-the-art tuners such as model-based optimization. For larger tasks, i.e. tasks with thousands of features of genetic data, parallelization of the benchmark experiments is mandatory. The package `BatchJobs` [67] and its successor `batchtools` [368] provide the bridge between `mlr` and managed high-performance computing clusters, allowing to compute comprehensive benchmarks on hundreds of CPUs simultaneously. In this way they can define and execute exhaustive benchmark studies, such as those from Lang, Kotthaus, Marwedel, Weihs, Rahnenführer, and Bischl [369], Lang [367] and Richter et al. [531].

The second framework extended for survival analysis is `mlr3` [370], the successor of `mlr`. The extension package `mlr3proba` [640] provides a general framework for probabilistic regression. Compared with the survival capabilities of `mlr`, `mlr3proba` connects much more learners and, even more importantly, connects and implements much more survival measures. Additionally, `mlr3proba` can be embedded in the infrastructure of the `mlr3pipelines` [65] package, which provides a language to build complex machine learning workflows as directed acyclic graphs. `mlr3pipelines` is also used to convert and unify the many predict types of survival models: while some models return a linear prediction vector, others return a continuous ranking, relative risks, or a complete time-dependent distribution such as individual survival function estimates. `mlr3proba` provides several pipeline operators for converting between predict types or even for composing multiple types.

Thanks to the unified interface of `mlr3` and `mlr3proba`, it is directly possible to use state-of-the-art methods to optimize the hyperparameters of the survival methods via `mlr3tuning`. Especially in the survival context, data preprocessing is often a crucial step. Decisions on how to configure the preprocessing should be included in this optimization process to obtain an unbiased estimate of the predictive performance. Modeling preprocessing through `mlr3pipelines` allows the building of a whole pipeline that can be resampled and optimized. To obtain an unbiased estimate of the performance of a pipeline identified through optimization, the whole optimization can be resampled, resulting in a nested resampling setting. Multi-criteria optimization is also supported, e.g. to tune for predictive performance, sparsity, and feature selection stability simultaneously by connecting the `stabm` [94] package.

2.5.7 Conclusion

The analysis of survival data requires the use of adequate statistical methodology, especially when it comes to accounting for missing information due to censoring. Corresponding methods are available and established. However, for modern high-dimensional data increasingly being generated today, omics data in particular, additional challenges emerge. Estimating prediction models often requires elaborate feature selection and hyperparameter optimization. For this task, Bayesian optimization meth-

ods provide a beneficial solution. They can efficiently identify models with competitive prediction accuracy out of a large set of candidate models. Of great importance is the availability and use of software frameworks for reproducible analysis pipelines. One valuable example is the widely used R package `mlr3`, which provides efficient, object-oriented programming on the building blocks of machine learning, together with its extension package `mlr3proba`, which provides a general framework for probabilistic regression, including many popular survival models and survival measures.

2.6 Protein Complex Similarity

Bianca K Stöcker
Till Schäfer
Petra Mutzel
Johannes Köster
Nils Kriege
Sven Rahmann

Abstract: Proteins have manifold functions in living cells, including structural integrity, transport, defense against pathogens, or message transmission, to name but a few. Recent advances in Machine Learning appear to have solved the protein folding problem, i.e., how to obtain the three-dimensional functional protein structure from the amino acid sequence of the protein. However, proteins rarely act alone, but instead perform their functions together with other proteins in so-called *protein complexes*. Quantifying the similarity between two protein complexes is essential for numerous applications, e.g., for database searches of complexes that are similar to a given input complex. While similarity measures have been extensively studied on single proteins and on protein families, there is little work on modeling and computing the similarity between protein complexes yet. Because protein complexes can be naturally modeled as graphs, graph similarity measures may be used, but these are often computationally hard to obtain and do not take typical properties of protein complexes into account. We introduce a parametric family of similarity measures based on Weisfeiler-Leman labeling see "The Weisfeiler-Leman Algorithm for Machine Learning with Graphs" in Section 4.2 in Volume 1. Based on simulated complexes, we show that the defined family of similarity measures is in good agreement with edit similarity, a similarity measure derived from graph edit distance, though it can be computed more efficiently. Moreover, in contrast to graph edit similarity, the proposed measures allow for an efficient similarity search in large volumes of protein complex data. It can therefore be used as a basis for large-scale machine learning applications.

2.6.1 Introduction

Proteins fulfill manifold tasks in living cells, but they rarely act alone. Indeed, most cellular functions are enabled only when proteins physically interact with other proteins, forming protein complexes. DNA transcription is a typical example, where RNA polymerase II, general transcription factors, cell type specific transcription regulators, and mediator proteins interact.

Understanding protein complex formation and function is one of the big challenges of cell biology, and is approached by both experimental techniques and computational modeling. While the constituent protein sequences can be obtained from the genome, the computational prediction of real protein complexes from protein interaction networks appears to be much more difficult [61, 643], and we presently face a lack of experimental datasets on verified real protein complexes. Fortunately, new experimental technologies such as high-resolution protein-protein docking are about to enhance our understanding of complexes significantly in the near future [348, 490].

When studying biological entities such as protein sequences or protein complexes, a fundamental task is to define a measure of similarity between two such entities. For protein sequences, there is a well-established theory based on scoring matrices and alignment scores [496]. For protein complexes, it appears that no systematic effort to quantify similarity has been made yet. The purpose of the present article is therefore to discuss the different options for defining a similarity measure on protein complexes and for proposing a reasonable and computationally tractable definition of protein complex similarity. Establishing a similarity measure is not only important fundamentally, but there are many immediate applications.

Database search In the *database search problem* we are given a query complex and a large collection (database) of complexes, and the task is to find the complexes in the database whose similarity to the query exceeds a given threshold.

Comparing predictions Several complex prediction methods predict putative complexes by locating dense regions in a protein interaction network [180, 272, 424, 497], and for comparing complexes predicted by different algorithms, it is of interest to compute a maximum-weight matching between the output of two algorithms, where the weighting is given by a similarity function.

Summarizing and clustering When simulating complex formation based on available knowledge such as possible interactions and interaction constraints, it is helpful to aggregate the simulation output to focus on frequently seen or typical complexes, ignoring small differences. Aggregation or clustering by similarity thereby reduces data size and complexity. Such a task requires quantifying the similarity between two protein complexes.

When there are tens of thousands of different complexes subject to pairwise comparison, a similarity measure must be efficiently computable.

This contribution is based on a conference paper by the authors [644], adapted by permission from Springer Nature; Copyright © 2019 Springer Nature Switzerland AG.

2.6.1.1 Models for Protein Complexes

We first discuss models for protein complexes at different levels of detail, namely the *set*, *multiset*, and *graph* models.

While intuition suggests that protein complexes can be naturally described as graphs with proteins as vertices and physical interactions as edges, there are in fact different ways to formally describe a protein complex. We start with a given set P of all proteins of an organism, the building blocks of the complexes.

Set In its most simple form, a protein complex can be defined as a set (in the mathematical sense, i.e., without multiplicities) of proteins, i.e., as a subset $\{p_1, p_2, \ldots, p_n\}$ of P. Sets neither capture the multiplicities nor the nature of the physical interactions between the constituent proteins of a complex. Some experimental techniques only give set-type information, and several existing databases only provide this type of information, e.g., the CORUM database [551].

Multiset Formally, a multiset is a function $C : P \to \mathbb{N}_0$ that assigns a multiplicity to each protein $p \in P$ with $C(p) = 0$ for proteins p that are not part of the complex. We also use the multiset notation $C = \{\!\{p_1, p_1, p_2\}\!\}$ to express that $C(p_1) = 2$, $C(p_2) = 1$ and $C(p) = 0$ for all other $p \in P$. Defining a protein complex as a multiset of proteins gives a more accurate representation of the complex, but still does not consider the interaction topology.

Graph To add more information, we can define a protein complex as an undirected graph $C = (V, E, \ell)$ with labeled vertices V, such that each vertex $v \in V$ represents a protein and hence has a label $\ell(v) \in P$, each edge $e \in E \subseteq V \times V$ represents a physical interaction between the corresponding proteins, such that E is symmetric and C is connected. The graph description provides the interaction topology. We call this representation a *protein complex graph* and define its *size* as $|C| := |V| + |E|$. In the following, we use the terms protein complex and protein complex graph interchangeably.

A remark is in order to avoid confusion with *protein interaction networks*. Our definition of the graph model of protein complexes is formally identical to the definition of protein interaction network. However, there are important differences. The protein complex graph represents an assembly composed of multiple proteins that physically bind and co-exist in one temporal and physical space, while a protein interaction network represents proteins that may interact at some point of time, where individual interaction time points may be different. Hence, the protein complex graphs considered in this work typically consist of only a few vertices and edges, while interaction networks are much larger.

For the set and multiset models, a similarity measure is readily given by the *Jaccard similarity* (see Section 2.6.2.1). For graphs, various techniques exist such as graph kernels [353] or graph matching [741]. A particularly intuitive approach is the *graph edit distance*, which has been proposed for pattern recognition tasks more than 30 years ago [574]. A graph edit distance between graphs C and C' measures the total costs of the edit operations required to transform C into C'. Defining similarity via graph edit operations appears natural, but has computational disadvantages, as the graph edit distance generalizes the classical maximum common subgraph problem [109], which

is NP-complete [220] and hard to approximate with given guarantees [311]. Therefore, a large number of heuristics for computing the graph edit distance without provable guarantees have been proposed. A particular successful class of heuristics derives the edit costs from the solution of a linear sum assignment problem [90, 352, 535]. Recently several elaborate exact algorithms for computing the graph edit distance have been proposed, but are still limited to small graphs [132, 239, 388]. The binary linear programming formulation of [388] allows researchers to compare graphs of moderate size using highly-optimized general purpose solvers. However, when we want to compare many complexes, evaluating the edit distance between all pairs becomes infeasible in practice.

We therefore propose an efficient alternative. We define a family of similarity measures on graphs using the Jaccard similarity, which can be efficiently computed and even more efficiently estimated with locality-sensitive hashing techniques. Taking the graph structure into account is achieved by the Weisfeiler-Leman labeling of the vertices [724], propagating vertex labels between neighbors. This approach is different from recent work that approximates and bounds the graph edit distance [534] and has the advantage of scaling better to large-scale studies.

We find that the Weisfeiler-Leman (WL) similarity approximates edit similarity well, but can be computed much more efficiently. In addition, in large-scale database searches for complexes similar to a given query complex, we obtain an additional speed-up by an order of magnitude when filtering for high WL similarity using min-hashing. Finally, we discuss limitations and possible extensions of this work.

2.6.2 Methods

Our goal is to define a similarity measure between protein complexes that captures not only the (multisets of the) constituent proteins, but also the interaction topology (graph structure). We introduce a parameterized family of similarity measures on protein complexes, which are based on multiset comparisons of vertex labels in the graph representation and take the local neighborhood of each protein into account by using Weisfeiler-Leman labels.

2.6.2.1 Jaccard Similarity of Sets and Multisets

To compare sets or multisets, Jaccard similarity coefficients are an established quantity.

Let $M \subseteq U$ and $M' \subseteq U$ be two subsets of a common universe U. Then the *Jaccard similarity* between M and M' is defined as

$$J_{\text{set}}(M, M') := \frac{|M \cap M'|}{|M \cup M'|} \in [0, 1]\,. \tag{2.20}$$

This definition is extended to multisets as follows. Recall that multisets M and M' are functions $U \to \mathbb{N}_0$, assigning multiplicities $M(o)$ and $M'(o)$ to each object $o \in U$. (The

set definition can be seen as the special case where the value set is only $\{0, 1\}$ instead of $\mathbb{N}_0$.) Then the *Jaccard similarity* between M and M' is defined as

$$J_{\text{multiset}}(M, M') := \frac{\sum_{o \in U} \min\{M(o), M'(o)\}}{\sum_{o \in U} \max\{M(o), M'(o)\}} \in [0, 1]\,. \tag{2.21}$$

The definition of J_{multiset} can be reduced to that of J_{set} by augmenting the element names with a running index in each multiset. For example,

$$J_{\text{multiset}}(\{\!\{A, A, B, C, C\}\!\}, \{\!\{A, C, C, C\}\!\}) = J_{\text{set}}(\{A_1, A_2, B_1, C_1, C_2\}, \{A_1, C_1, C_2, C_3\})\,.$$

Using this transformation, sketching techniques like min-hashing [105] that primarily work on sets can be extended to multisets.

2.6.2.2 A Parametric Family of Protein Complex Similarity Measures

Instead of comparing (protein complex) graphs directly by their labels and graph topology, we extract and compare multisets of derived features that represent local neighborhood information. Encoding the local structure surrounding a vertex is a general method widely used in graph matching and machine learning with graphs. Various concepts and techniques for this have been proposed, e.g., the k-hop neighborhood [319, 459], or the k-sphere neighborhood [37]. Weisfeiler and Leman developed an iterative label refinement procedure to derive a canonical graph representation for graph isomorphism testing [724]. The same procedure is often used to define graph similarities or graph kernels [602]. This approach recently became popular in machine learning for its expressivity and favorable algorithmic properties. Several highly efficient graph kernels based on Weisfeiler-Leman refinement have been proposed [351, 602], and several graph neural networks were shown to be at most as expressive as the Weisfeiler-Leman method [737].

The technique works as follows. Initially, the feature multiset of a graph consists of the union of all vertex labels, i.e., the protein names. After the initialization, the vertex labels are iteratively augmented by the labels of the neighboring vertices from the previous iteration, thereby encoding the (local) graph structure in the vertex labels. As previously mentioned, we label the vertices of graphs with the protein names; so two vertices that refer to the same protein type are labeled identically. Thus, the initial feature multiset is identical to the multiset representation as described above. Let us now formally define the process.

Definition 1 (Weisfeiler-Leman labeling)**.** Let $C = (V, E, \ell_0)$ be a graph with label function $\ell_0 : V \to L_0 := P$, where P is the initial set of vertex labels (e.g., protein names for protein complex graphs). Furthermore, let $N(v) := \{u \mid \{v, u\} \in E\}$ denote the neighbors of vertex $v \in V$. Then, the Weisfeiler-Leman labeling of iteration i is defined as a re-labeling of the graph. It replaces the current labeling function $\ell_{i-1} : V \to L_{i-1}$ with a new labeling function $\ell_i : V \to L_i$. The value of ℓ_i for a vertex $v \in V$ is recursively

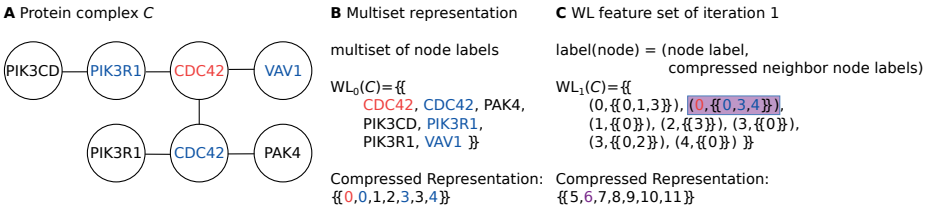


Fig. 2.27: Example of a protein complex and its representations. The colors highlight the labels of an example node in $\mathrm{WL}_0(C)$ and $\mathrm{WL}_1(C)$. **A:** Graph representation of protein complex C. **B:** Multiset representation of C which is equal to $\mathrm{WL}_0(C)$. **C:** Result of the first WL iteration.

defined as

$$\ell_i(v) := (\,\ell_{i-1}(v),\ \{\!\{\ell_{i-1}(u) \mid u \in N(v)\}\!\}\,). \tag{2.22}$$

Note that the second component of the new label is a multiset.

To avoid increasingly complex labels consisting of nested multisets, label compression is performed after each step. This is achieved by applying an injective function, which maps a pair consisting of a label and a multiset of labels of the form given in Equation 2.22 to an integer label. The label compression step must be consistent across multiple graphs in order to construct comparable feature sets. If all graphs in the dataset are known from the beginning, we can sort all the multisets of one iteration, identify the identical pairs, and assign them to the same new integer label. This step can be realized in time linear in the total number of edges of all graphs by applying variants of bucket sort [602]. A less efficient, but more flexible approach, which is suitable even when the graphs are only revealed successively, is to manage the injective map used for label compression in a hash table.

Given the Weisfeiler-Leman labeling function for any iteration i, we can now define the multiset of Weisfeiler-Leman features for iteration i.

Definition 2 (Weisfeiler-Leman feature set)**.** Let $C = (V, E, \ell_0)$ be a graph with label function $\ell_0 : V \to L_0 = P$, where P is the initial set of vertex labels. Then, the Weisfeiler-Leman features of iteration i are defined as multiset $\mathrm{WL}_i(C) = \{\!\{\ell_i(v) \mid v \in V\}\!\}$.

Note that $\mathrm{WL}_0(C)$ always corresponds to the initial multiset of labels (protein names). Accordingly, $\mathrm{WL}_1(C)$ integrates the neighborhood labels of each node. Figure 2.27 shows an example protein complex graph, together with the associated feature sets $\mathrm{WL}_0(C)$ and $\mathrm{WL}_1(C)$. A node and its neighborhood are highlighted in red and blue to demonstrate the relation between $\mathrm{WL}_0(C)$ and $\mathrm{WL}_1(C)$.

We use the Jaccard coefficient to obtain a normalized similarity based on multiset intersection. We apply the Jaccard coefficient to the feature sets of each iteration individually and compute a convex combination of the results. Let $w = (w_i)_{i\geq 0}$ be a sequence of non-negative weights with $\sum_{i\geq 0} w_i = 1$. We quantify the weighted similarity between

two labeled graphs C and C' by

$$S_w(C, C') := \sum_{i \geq 0} w_i \cdot J_{\text{multiset}}(\text{WL}_i(C), \text{WL}_i(C')), \tag{2.23}$$

where J_{multiset} is given by Equation 2.21. This defines a family of similarity measures between labeled graphs with values in $[0, 1]$, parameterized by the weight vector $w = (w_0, w_1, \dots)$.

It is easy to see that, as long as $w_0 > 0$, we have $S_w(C, C') = 0$ if and only if the label sets of C and C' are disjoint. If $S_w(C, C') < 1$, the graphs are not isomorphic. However, $S_w(C, C') = 1$ does not necessarily imply that C and C' are isomorphic even if $w_i > 0$ for all i. There exist examples of non-isomorphic graphs G, G' with $\text{WL}_i(G) = \text{WL}_i(G')$ for all $i \geq 0$. (As a simple example, take G to be a cycle of six vertices, and G' to be two cycles of three vertices, all with the same label.) On the other hand, there exist classes of graphs, such as the CR-graphs, for which the implication "$S_w(C, C') = 1 \Rightarrow C, C'$ are isomorphic" is true if $w_i > 0$ for all i [31]. Moreover, the implication holds with high probability for random graphs (without vertex labels) even when $w_i = 0$ for all $i \geq 3$ [34].

In our application scenario, we may assume that most protein complexes are non-adversarial graphs with sufficiently simple structure and expressive initial labels such that their Weisfeiler-Leman features are appropriate to characterize their similarity. In fact, we put forward the hypothesis that using a single iteration is frequently sufficient for practical purposes, and we set $w_i := 0$ for $i \geq 2$ in our computational experiments (see Results) and only have a single free parameter $w_1 \in [0, 1]$ that defines $w_0 := 1 - w_1$. In this case, S_{w_1} is efficiently computable: a proof of the following lemma can be found in the work of [602].

Lemma 3. *For $w_1 \in [0, 1]$, each of the one-parameter similarity measures*

$$\begin{aligned} S_{w_1}(C, C') := (1 - w_1) \cdot J_{multiset}(WL_0(C), WL_0(C')) \\ + w_1 \cdot J_{multiset}(WL_1(C), WL_1(C')) \end{aligned}$$

can be computed in $O(|C| + |C'|)$ time, where $|C| = |V| + |E|$.

2.6.2.3 A Similarity Measure Based on Graph Edit Distance

To compare the family of Weisfeiler-Leman multiset-based similarity measures defined above with graph edit distance, we state a formal definition of the edit-based similarity. We allow the following elementary operations to edit a graph: vertex deletion, vertex insertion, vertex relabeling, edge deletion, and edge insertion. A sequence $(o_1, \dots, o_k)$ of such edit operations that transforms a graph G into another graph H is called an *edit path* from G to H. Each operation o is assigned a cost $c(o)$, which is zero for substituting vertices and edges with the same label. We use a cost of 1 for all operations except vertex relabeling, which has a cost of 2, corresponding to one deletion and one insertion (leaving the edges in place). Note that deleting or inserting a vertex of degree k otherwise

has cost $k+1$ for deleting k edges and the vertex itself. We denote the set of all possible edit paths from G to H by $Y(G,H)$.

Definition 4. Let G and H be labeled graphs. The *graph edit distance* from G to H is defined by

$$d(G,H) = \min\left\{\sum_{i=1}^{k} c(o_i) \mid (o_1,\ldots,o_k) \in Y(G,H)\right\}. \tag{2.24}$$

Intuitively, the graph edit distance preserves a subgraph G' of G that is also contained in H using zero-cost substitutions, deletes the vertices and edges in G that are not in G', and then inserts vertices and edges to obtain an isomorphic copy of H. Therefore all non-zero costs can be attributed to the elements that are in one of the graphs, but not in their common subgraph. In this sense the graph edit distance is similar to the symmetric difference of two sets. This observation motivates the following normalized similarity measure derived from the graph edit distance. We define the *graph edit similarity* as

$$S_g(G,H) := \frac{|G|+|H|-d(G,H)}{|G|+|H|+d(G,H)} \in [0,1], \tag{2.25}$$

where $|G| := |V(G)| + |E(G)|$. Note that the graph edit distance between G and H is at most $|G|+|H|$, which is achieved by deleting all vertices and edges of G and inserting all vertices and edges of H. In this case the graph edit similarity is zero. Similarly, $S_g(G,H) = 1$ if and only if $d(G,H) = 0$. In this respect the similarity measure resembles the Jaccard similarity. In fact, the following lemma shows that, if the edges are not taken into account, the graph edit similarity equals the multiset Jaccard similarity. Therefore, the graph edit similarity can indeed be seen as a natural extension of the multiset Jaccard similarity to graph structured data.

Lemma 5. *For two vertex-labeled graphs G, H, let C, D denote their respective label multiset. For the edge-free graphs $G' = (V(G),\emptyset)$ and $H' = (V(H),\emptyset)$ it holds that $S_g(G',H') = J_{multiset}(C,D)$.*

Proof. An optimal graph edit path is obtained as follows: We substitute the vertices with common labels free of cost, which are $Z = \sum_{p\in P} \min\{C(p), D(p)\}$ in total. We delete the remaining $|G'| - Z$ vertices in G' and insert $|H'| - Z$ vertices to obtain an isomorphic copy of H' at a total cost of $|G'| + |H'| - 2Z = d(G,H)$. Instead we may also substitute up to $||G'| - |H'||$ vertices, each at cost two, which results in the same total cost. Using the fact that $|G'| = \sum_{p\in P} C(p)$ and $|H'| = \sum_{p\in P} D(p)$, we obtain the result

by calculating

$$S_{\mathrm{g}}(G', H') = \frac{|G'| + |H'| - d(G', H')}{|G'| + |H'| + d(G', H')} = \frac{Z}{|G'| + |H'| - Z}$$

$$= \frac{Z}{\sum_{p \in P} C(p) + \sum_{p \in P} D(p) - Z}$$

$$= \frac{\sum_{p \in P} \min\{C(p), D(p)\}}{\sum_{p \in P} C(p) + D(p) - \min\{C(p), D(p)\}}$$

$$= \frac{\sum_{p \in P} \min\{C(p), D(p)\}}{\sum_{p \in P} \max\{C(p), D(p)\}} = J_{\mathrm{multiset}}(C, D)\,.$$

□

2.6.2.4 Database Searches

An important application of a similarity measure $S(\cdot, \cdot)$ is for similarity searches in large databases. When searching for similar protein complexes to a given input complex ("query" Q) in a large database, one can perform a linear scan and compute $S(Q, C)$ for each complex C in the database and report those with $S(Q, C) \geq T$ for a given threshold T. However, computing $S(Q, C)$ exactly may be computationally expensive, and in many cases, a quickly computable upper or lower bound can be used as an initial filter.

In Section 2.6.3, we evaluate the proposed similarity measure S_{w_1} against graph edit similarity S_{g}, and for database searches we use individual filtering techniques as described in this section.

Weisfeiler-Leman Similarity For S_w, a speed-up is possible using min-hashing [105], which is a locality-sensitive hashing scheme for the Jaccard similarity between sets that can be extended to multisets, as described in Section 2.6.2.1. We use a simple approach that maps the $\mathrm{WL}_i(C)$ multisets to integer hash values $h_{i,j}(C)$ using a large number ($j = 1, \ldots, K$) of different random hash functions. The exact $S_w(Q, C)$ value is only computed if any of the hash values agrees with that of the query, i.e., if $h_{i,j}(C) = h_{i,j}(Q)$ for any $i = 0, 1$ and $j = 1, \ldots, K$. The number K of hash functions is chosen such that the false negative error rate is lower than a given probability threshold (0.01).

Graph Edit Similarity The graph edit *distance* is widely used for searching graph databases and several approaches tailored to this task have been proposed [132, 323, 397, 712, 755, 768, 769, 770]. These methods typically follow a *filter-verification* approach [755]. In the filter phase, efficiently computable lower bounds on the graph edit distance are used to eliminate dissimilar graphs. The remaining graphs are then verified using upper bounds on the graph edit distance and, if necessary, by an exact algorithm to obtain the final result. The methods can be categorized according to different criteria.

Several methods compute a lower bound for every graph in the database [132, 323, 755], while the others use an index data structure to find candidates without scanning the whole database. The lower bounds are often derived from overlapping substructures, while some methods partition the graphs into disjoint parts [323, 397, 769]. Recently, the large number of proposed lower and upper bounds were systematically studied in an extensive experimental evaluation according to their running time and tightness [90].

Most of the above mentioned algorithms assume a uniform edit cost function and cannot directly be applied to solve database search problems with respect to the graph edit similarity as defined in Equation 2.25: To make graph similarity compatible with the Jaccard index, we use a cost of 2 instead of 1 for vertex relabeling. Nevertheless, several of the efficiently computable bounds for uniform graph edit distance can be generalized for this case, and we have implemented such generalizations (see below). Then, by substituting $d(G, H)$ in Equation 2.25 by known lower (upper) bounds on the graph edit distance, we obtain upper (lower) bounds for the graph edit similarity.

Since the vertex labels denoting proteins are highly specific when comparing protein complexes, we derive a first lower bound on the graph edit distance from the cardinality of the symmetric difference of the two vertex label multisets [323, 768]. This is equivalent to using J_{multiset} as an upper bound for the graph edit similarity.

As a second lower bound we use a more expensive approach based on the linear sum assignment problem, which was shown to provide a good trade-off between tightness and running time [90]. The assignment instance is defined on the vertices of the two graphs, where additional dummy vertices are introduced that represent vertex insertion and deletion. The costs for assigning individual vertices is made up of the costs for substituting the vertex label and the cost of an optimal assignment between the incident edges [535]. Since we do not consider edge labels, the assignment cost matrix can be computed in quadratic time (cf. heuristic `BRANCH-CONST` in [90]) and the instance can be solved in cubic time. The cost of the assignment instance serves as a lower bound on the graph edit distance. Following [535], we obtain an upper bound from the cost of the edit path derived from the assignment. If this is not sufficient for verification, we compute the exact graph edit distance using the binary linear programming formulation of [388].

2.6.3 Results

2.6.3.1 Hypothesis

We hypothesize that finite truncations of the Weisfeiler-Leman-based family of similarity measures S_w (defined in Equation 2.23), in particular S_{w_1} (Lemma 3), are in good agreement with the edit similarity (defined in Equation 2.25) for typical protein complex graphs. Especially S_{w_1} has the advantage that it can be efficiently computed.

2.6.3.2 Experimental Setup

We have implemented the similarity measures based on Weisfeiler-Leman labeling and the graph edit similarity in Java 8. To compute the exact graph edit distance, we used a recent binary linear programming formulation [388] and solved the instances using Gurobi 7.5.2. All experiments were run on 18-core Intel Xeon E5-2699 CPUs with 2.3 GHz and 512 GB RAM, using 64-bit Ubuntu Linux 18.04. Our analysis is available as a Snakemake workflow [345] on github (https://github.com/BiancaStoecker/complex-similarity-evaluation).

2.6.3.3 Data Generation

As mentioned in the introduction (Section 2.6.1), obtaining graphs from real protein complexes is difficult at the moment, because experimental techniques that resolve the (graph) topology of the complexes are still in the developmental stage. Therefore we resort to the simulation of complexes, based on two types of knowledge: possible physical protein-protein interactions, formalized by a *protein interaction network*, and *constraints between protein interactions*. Especially the second type of information allows us to simulate more realistic complexes than what we would get from interaction networks alone.

Formally, a protein interaction network is an undirected graph $N = (P, I)$, where P is the set of protein types of a cell (or an organism), and $I \subset P \times P$ indicates the pairs of protein types that may potentially physically interact. Since N describes the entirety of possible interactions, any protein complex can be seen as a connected subgraph of N.

Protein interactions are not independent of each other, but interdependent. Those interaction dependencies are generated by two major mechanisms: allosteric regulation, in which the capability of a protein to bind other proteins is affected by a conformational change upon one interaction [374], and steric hindrance, which prevents proteins from binding to identical or nearly identical protein domains, leading to mutual exclusiveness of interactions [573]. The dependencies between interactions constrain the set of possible protein complexes and their assembly paths. One possible model for this are *constrained protein interaction networks*, where the protein interaction network is enhanced by the interaction dependencies (*constraints*) modeled as propositional logic formulas [649]. With constrained protein interaction networks, we can stochastically simulate complex formation based on the available knowledge and obtain a detailed interaction topology (which proteins physically interact) for each complex.

For the simulation, a *constrained protein interaction network* was generated from the human adhesome protein network and a set of interaction dependencies obtained from protein domain interaction databases and manual curation (for details, see [649]). Then, protein complex assembly was simulated in a step-wise process, with association and dissociation rates calibrated to fit the complex size distribution of the CORUM database [551], until reaching convergence. The obtained complexes mimic the size distribution of known complexes, while also providing information about the actual

Tab. 2.5: Properties of the 2 242 972 simulated protein complex graphs. There are 717 distinct vertex labels (protein names), the most frequent ones being 'GRB2' occurring 100 009 times, EGFR occurring 84 706 times, and 'CRK' occurring 83 117 times. The least frequent labels are 'PHF20' (55×), 'CDYL' (63×), and 'PHF20L1' (75×). The average label frequency is 14561.6. The right table shows how the number of distinct labels increases with WL iterations; here $|\mathrm{WL}_{\leq i}|$ refers to the cardinality of the set of all labels up to the i-th WL iteration.

Quantity	Average	Min	Max
$\lvert V\rvert$	4.655	3	126
$\lvert E\rvert$	3.655	2	125
Degree	1.570	1	28
Diameter	2.709	2	28
Density	0.530	0.016	0.667

Label set sizes	
$\lvert \mathrm{WL}_0\rvert$	717
$\lvert \mathrm{WL}_{\leq 1}\rvert$	461 657
$\lvert \mathrm{WL}_{\leq 2}\rvert$	4 114 569
$\lvert \mathrm{WL}_{\leq 3}\rvert$	9 237 071
$\lvert \mathrm{WL}_{\leq 4}\rvert$	14 703 231

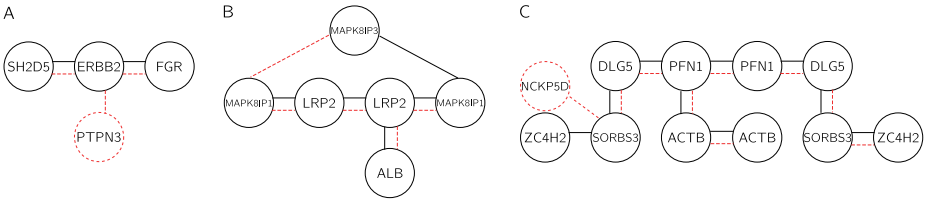


Fig. 2.28: Three exemplary pairs of protein complexes. Each labeled node is a protein instance, each edge a protein interaction, and solid black vs. dashed red edges distinguish between the two complexes. **A:** Edit similarity 0.714; WL similarity in [0.4, 0.75] depending on weight w_1. **B:** Edit similarity 0.838; WL similarity 1.0 (independent of w_1). **C:** Edit similarity 0.9; WL similarity in [0.667, 0.818] depending on w_1.

physical interactions happening inside the complex, an information that is currently not yet available for real data. Over 2.2 million protein complex graphs were simulated in this way. Some statistics are given in Table 2.5: Most simulated graphs are small and tree-like ($|V| = |E| + 1$) and consist of low-degree nodes. We were able to verify that all distinct simulated graphs can be distinguished by WL labels after at most two iterations.

To evaluate the Weisfeiler-Leman based similarity ("WL similarity") against the edit distance based similarity ("edit similarity"), we computed both measures on selected pairs of simulated complexes.

2.6.3.4 Illustrating examples

We first consider three exemplary pairs (Figure 2.28 A–C) with edit similarities of approximately 0.7, 0.8 and 0.9, respectively, the latter being the most similar observed pair. Our simulation has been calibrated to yield complexes of a realistic size distribution that additionally reflect all currently known interaction dependencies. However, since this data is likely incomplete and we also did not consider the law of mass action, we

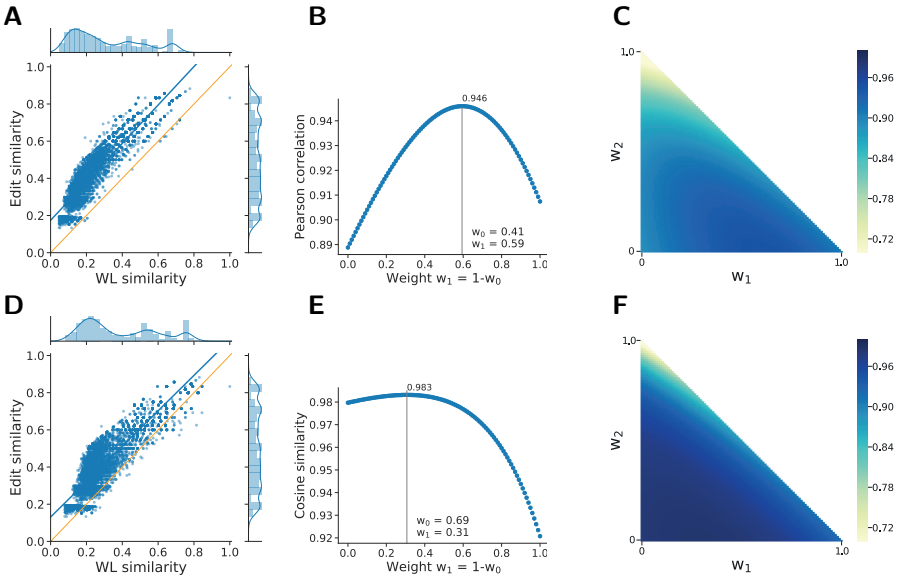


Fig. 2.29: Comparison of edit similarity and WL similarity. A and **D:** Scatterplot between edit similarity and WL similarity for weight $w_1 = 0.59$ with maximum Pearson correlation (A) and $w_1 = 0.31$ with maximum cosine similarity (D), including marginal distributions and least-squares regression line. Each point represents a pair of complexes. **B:** Pearson correlation coefficient between edit and WL similarity as function of w_1. The maximum 0.946 occurs for $w_1 = 0.59$ (scatterplot A). **E:** Cosine similarity as a function of w_1. The maximum 0.983 occurs for $w_1 = 0.31$ (scatterplot D). **C** and **F:** Heatmap showing the Pearson correlation (C) and cosine similarity (F) over weights w_1 and w_2 when using 2 WL iterations and $w_0 = 1 - (w_1 + w_2) \geq 0$.

do not claim that the particular combination of proteins in these examples is likely to occur in reality. The examples are therefore only meant to illustrate the behavior of the two measures and give an intuition of cases where WL similarity fails to properly approximate edit similarity.

In example A, an additional protein (PTPN3) is added to an existing complex, a linear chain of 3 proteins. The edit similarity is $10/14 = 0.714$, the WL similarity is between 0.75 for $w_1 = 0$ and 0.4 for $w_1 = 1$. Because the edit similarity is between the extreme WL similarities, there exists a unique weight $w_1^* \approx 0.102$, for which WL and edit similarities agree for this particular complex pair. Example B is a noteworthy case, because the WL similarity is 1.0, independently of w_1, because the vertex labels are identical even after the first Weisfeiler-Leman iteration. (Further iterations would show a difference.) The edit similarity is $20/24 = 0.83$, which is obtained by attaching ALB to the other LRP2 protein. In example *C*, one protein is replaced by another one in a fairly large complex. The edit similarity (0.905) is relatively high and outside the WL similarity range between 0.667 for $w_1 = 1$ and 0.818 for $w_1 = 0$.

2.6.3.5 Large-Scale Comparison

For the following comparison, we considered only a stratified subset of all possible pairs for the analysis, because calculating the edit similarity is computationally costly. To obtain candidate pairs, we considered all pairs of complexes that have at most 20 proteins (larger complexes are so rare that high similarities are unlikely), that have a size difference of protein multisets of at most 10, and that share at least one protein. These were sorted in descending order according to the number of shared proteins. Then the edit similarity was computed on the first 500 000 candidate pairs, and the similarity values were grouped into bins of width 0.1. Because most pairs of complexes share a small number of proteins, we find many pairs with small edit similarity (but none in the range [0.0, 0.1[because we required one common protein) and fewer pairs with edit similarity above 0.5. To achieve a uniform distribution among bins for the comparison, we randomly selected 1000 pairs from each bin, excluding the bin [0.9, 1.0[which contained a single pair. This yielded 8000 pairs of complexes from 8 bins.

Because most protein complexes are small and do not exhibit properties of examples B or C of Figure 2.29, the overall agreement between WL similarity and edit similarity is high. For each of the selected complex pairs, we computed the exact edit similarity and the WL similarity for each weight $w_1 \in W := \{0.0, 0.01, \ldots, 1.0\}$ and $w_0 := 1 - w_1$. Let e be the vector of edit similarity values and $s(w_1)$ the corresponding vector of WL similarity values using weight w_1 for WL in the first WL iteration. To compare the similarity measures, we calculated both the Pearson correlation coefficient and the cosine similarity of e and $s(w_1)$ for all $w_1 \in W$. As can be seen from Figure 2.29B, the highest Pearson correlation values occur for w_1 between 0.56 and 0.62. The maximum Pearson correlation coefficient of 0.946 is obtained for $w_1 = 0.59$. Figure 2.29A shows the scatter plot between the similarities for this weight. For the cosine similarity, the maximum value is reached for weight $w_1 = 0.31$, but the function is less peaked, and all values of $w_1 < 0.6$ lead to high agreement (Figure 2.29E).

To quantify the possible benefit of additional WL iterations (and hence a larger space of possible weight vectors), we first repeated the calculations with an additional second WL iteration. We calculated the similarity measure for all weight combinations $(w_1, w_2) \in W \times W$ with $w_0 := 1 - (w_1 + w_2) \geq 0$ and $w_0 + w_1 + w_2 = 1$. The Pearson correlation and cosine similarity over the weights w_1 and w_2 are shown in Figure 2.29 C and F. The weight combinations that lead to maximum correlations all have $w_2 = 0$ and therefore are the same as in the similarity measure without second iteration (Figure 2.29 A and D). Therefore, the additional WL iteration provided no benefit for approximating edit similarity.

Overall, we find good agreement between edit similarity and WL similarity for appropriate values of w_1; values around $w_1 = 0.5$ yield both high Pearson correlation and cosine similarity.

2.6.3.6 Runtime Comparison

To study the practical running time of both similarity measures, two different settings were evaluated. In the first setting, all pairwise similarities in a subset of the dataset were computed. This setting is a typical sub-task of clustering or machine learning tasks on metric distances. In the second setting, the database search application was evaluated. Given a set of query complexes and a similarity threshold, similar complexes in the dataset were searched.

Pairwise Similarities We measured the time required to compute all 10 000 pairwise similarities for a subset of 100 graphs, drawn at random from the dataset described in Section 2.6.3.3. For the Weisfeiler Leman-based similarities, the computation can be divided into two steps. In the first step, the Weisfeiler-Leman feature vectors are computed for each of the 100 graphs (WL-FV). In the second step, they are used to compute the similarity values between all 10 000 pairs (WL-SIM). Thus, the computational costly part is computed only once for each graph, while the quadratic number of comparisons is lightweight. This kind of preprocessing is not possible for the graph edit similarity, where each pairwise similarity computation is costly (GES).

Figure 2.30 shows the violin plot of the measured times for each single feature vector and pairwise similarity calculation. Running times are shown separately for one and two WL iterations (WL-[FV|SIM]-[1|2]; GES). As expected, the calculation with two WL iterations is slower than using only one iteration, but the difference is small. Most importantly, we observe that a single graph edit similarity computation is two to four orders of magnitude slower than a single Weisfeiler Leman-based similarity computation (median of over 10^6 vs. approx. 10^3 ns, but extreme outliers are visible from the violin plots). While Figure 2.30 shows the times for single-instance calculations, Table 2.6 shows the total times for all 100 (FV) and 10 000 (SIM) calculations. For Weisfeiler-Leman similarity, we observe that computing the feature sets dominates the running time. (This eventually changes if more graphs are considered since the WL-FV time grows linearly but the WL-SIM time grows quadratically with the size of the dataset.) Overall, the GES computation is slower by more than four orders of magnitude, which is influenced by a few extremely slow GES computations.

Database Search We used the entire set of graphs described in Section 2.6.3.3 as a database and 500 randomly selected graphs as queries. For the Weisfeiler Leman-based similarities, we evaluated a linear scan over the database (`wl_linear`) and the min-hashing speed-up described in Section 2.6.2.4 (`wl_minhash`) with false negative rate 0.01 and weights $w_0 = w_1 = 1/2$. Times were measured separately for the Weisfeiler-Leman feature set calculation, the hash table creation (for `wl_minhash` only), and the queries itself. For the graph edit similarity (`ges`), a linear scan over the database is prohibitive and hashing schemes are not readily available. Therefore, we employed the filter-verification approach described in Section 2.6.2.4 and measured the time for filtering

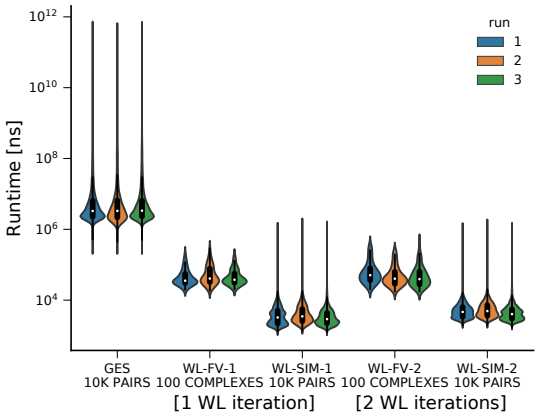


Fig. 2.30: Running times. Violin plots of wall-clock times for computing each of the 10 000 pairwise graph edit similarities (GES), the 100 Weisfeiler-Leman sets (WL-FV) and their 10 000 pairwise Jaccard coefficients (WL-SIM) with either one or two Weisfeiler-Leman iteration(s) in three independent runs. The time axis is logarithmic.

Tab. 2.6: Total running times for each of three runs in seconds. WL-Total = sum(WL-FV) + sum(WL-SIM) for either 1 or 2 Weisfeiler-Leman iteration(s). Here sum() refers to the sum over the 100 feature vector calculations (WL-FV) and 10 000 similarity calculations (WL-SIM) whose time distribution is depicted in Figure 2.30.

Time [s]	Run 1	Run 2	Run 3
GES	851.222	809.087	845.883
WL-Total-1	0.050	0.056	0.045
WL-Total-2	0.066	0.071	0.055
sum(WL-FV-1)	0.005	0.006	0.005
sum(WL-SIM-1)	0.045	0.050	0.040
sum(WL-FV-2)	0.008	0.006	0.006
sum(WL-SIM-2)	0.058	0.065	0.049

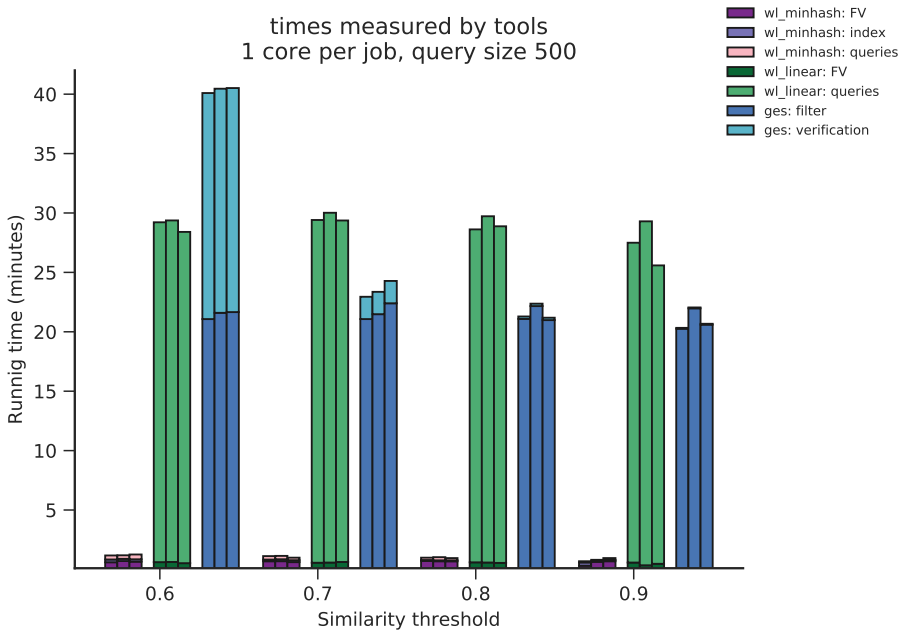


Fig. 2.31: Running time comparison of the three approaches wl_minhash (purple), wl_linear (green), and ges (blue) for a database search with 500 queries and a database of ≈ 2 million complexes.

and verification separately. All three algorithms were run with the similarity thresholds 0.6, 0.7, 0.8 and 0.9 and each run was repeated three times. Further, each run was executed using only a single CPU core for better comparability of CPU time usage; in practice, many database complexes can be evaluated in parallel, independently of each other. The results are shown in Figure 2.31.

It can be seen that `wl_minhash` and `wl_linear` need a similar amount of time for the feature vector calculation, but the query time is much smaller for `wl_minhash` because only a small fraction of the database complexes needs to be evaluated. These savings outweigh by far the additional indexing time required to build the hash tables, even for only 500 queries. Therefore, `wl_minhash` will scale even better to larger numbers of queries. The filter-verification approach of the graph edit similarity `(ges)` is slightly faster than `wl_linear` for thresholds 0.7, 0.8, and 0.9, because using the similarity bounds often suffices for a decision and the actual graph edit distance does not need to be computed often. For threshold 0.6, however, many verifications and hence exact computations are necessary; so the running time for ges increases above that of `wl_linear`.

2.6.4 Discussion

Our original motivation to consider protein complex similarity was to reduce the size of the simulation output of our constrained protein interaction network simulator [649] via clustering, and we were surprised to see that apparently, no similarity measures have been proposed explicitly for protein complexes in the literature. Depending on the underlying representation (set, multiset, or graph), different alternatives suggest themselves. However, most graph-based measures are both theoretically and practically hard to compute for larger complexes or for large amounts of complexes. While different tractable graph similarity measures [705] and an approximate graph edit distance [534] have been developed, none of these appear to be specifically tailored to the properties of protein complexes (often less than ten vertices; sparse).

Our proposal to define the similarity as a convex combination of two Jaccard coefficients (protein label multiset and Weisfeiler-Leman label multiset after one iteration) has several beneficial properties. We have shown that it can approximate edit-based similarity with high Pearson correlation and cosine similarity, and at the same time, that it can be computed much more efficiently. Further, for weight $w_0 = 1$ of the 0-th WL iteration, WL similarity reduces to the natural similarity measure of the multiset representation. Our framework hence allows for a smooth transition between multiset and graph representation. The comparison with an edit-based similarity seems to indicate that the protein label multiset plays an important role if one wants to approximate the edit similarity, the first WL iteration provides additional graph information, and the second WL iteration does not provide further benefits, probably because most complexes consist of few proteins. In addition, in large-scale similarity searches, using Jaccard coefficients allows us to efficiently pre-filter for high similarity using locality-sensitive hashing. In combination with the preprocessing abilities discussed in the experimental running time comparison, this allows for very fast search queries, clusterings, and other applications that rely on intensive distance computations.

In the present work, we have not considered individual similarity between vertex labels (i.e., protein types): We treat two labels as either equal or distinct. While it is relatively straightforward to allow arbitrary label similarities in the graph edit distance framework, and, via Equation 2.25, in the graph similarity framework, this generalization appears less straightforward for WL similarity and will be investigated in future work.

From a biological point of view, a high similarity value between two protein complex graphs should indicate a high probability that the complexes share biological functions and can (partially) substitute each other in a cellular process. If comprehensive functional information were available, we could use it for evaluating different similarity measures and decide which ones best capture the biological reality. At present, when not even the interaction topology of most complexes has been determined, such an evaluation is unfeasible, but this may change in the future.

3 Industry 4.0

3.1 Keynote on Industry 4.0

Michael ten Hompel
Moritz Roidl

Abstract: Industry 4.0 is the connecting element of all contributions in this chapter. The notion of a physical thing moving within an industrial process unites the research presented. Logistics is a driving force behind Industry 4.0 developments and that the concept of a cyber-physical twin is one of its fundamental building blocks. This keynote introduces and develops the mindset for fully appreciating the following contributions. They range from the creation of a steelbar to autonomous swarms of drones, providing a wide-ranging selection of current developments.

3.1.1 Introduction

The term "Industry 4.0" heralds a fourth industrial revolution. The factory of the future is built upon a hyper-connected, smart, and autonomous infrastructure. It promises to deliver high adaptability with optimal use of resources. Industry 4.0 is expected to bring considerable benefits for production sites, factory equipment suppliers, and business software providers, on both the productivity and the revenue side. Under Industry 4.0, digitization penetrates all areas of industrial process chains–from production to distribution to recycling and waste management–and is thus spurring expectations and ideas regarding the future design and implementation of processes, facilities, and systems.

3.1.2 Industry 4.0

Amid the wide variety of views on the term Industry 4.0, this keynote adheres to the German concept of a Plattform Industrie 4.0. According to this, the term Industry 4.0 "stands for the fourth industrial revolution, a new stage in the organization and control of the entire value chain" and is intended to establish a link to the three previous industrial revolutions. Many authors refer to Industry 4.0 collectively as the digital transformation of the manufacturing sector. A comprehensive overview of the field is given in [279], [548], and [697].

As the term revolution suggests, the transformation is likely to present a variety of opportunities for an economy that can deal with the disruption that comes with

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.
https://doi.org/10.1515/9783110785982-003

profound change. In addition, the successful implementation of Industry 4.0 concepts and technologies requires the ability to easily change business processes with regard to new requirements. This sense of a high adaptability calls for the creation of a kind of "deep transparency" to efficiently evaluate decision-making situations. The basic vision of Industry 4.0 thus stems from the age-old desire for the constant availability of all relevant information about all physical objects at all levels of industrial process chains. Technologically, this "deep transparency" is to be achieved through the massive use of low-cost sensor technology. To ensure constant availability of information, state-of-the-art communication technologies provide the means for both networking the production technology and networking between the hierarchical levels of the IT architecture. From a business perspective, the hope is that this comprehensive networking will give rise to highly adaptable supply-chain networks that can organize and optimize themselves and thereby also form the basis for a wide range of business model innovations (cf. [372], [189]).

Concepts

Industry 4.0 encompasses various concepts that not only cover different technologies, but also affect various structural aspects of the organization within a company and between companies.

The constant availability of information requires a smooth exchange of data within the entire value creation network through the integration of physical objects and IT systems. This gives rise to the concepts of *vertical and horizontal integration*. Vertical integration describes the creation of a coherent network for objects and systems within a company. All internal IT systems are interconnected via harmonized interfaces that serve to exchange data between a single sensor, a production machine, or the production planning system. Horizontal integration takes place across the entire value creation network. The vertically integrated IT systems of customers, suppliers, or the company's own distributed sites are integrated into a horizontal system landscape. This enables the exchange of information in real-time across company boundaries (cf. [310], [548]).

While the goal of constant availability of information can theoretically be achieved through a centralized approach, Industry 4.0 explicitly recognizes the geographically distributed nature of the value creation network. This is covered by the concepts of *decentralized control and autonomous behavior* of physical objects. These concepts encompass two important aspects. On the one hand, the disruptive dissolution of centralized, rigidly planned control systems is necessary simply from the point of view of transmission and computing power due to the large amounts data that are to be delivered in real-time. On the other hand, decentralized control enables the use of autonomous, automated decision support systems. Based on the available data, these systems should support human decisions or make them partially autonomous (cf. [617]).

The digitization of all physical objects is reflected in the concept of Cyber-Physical Systems (CPS). These are created by combining a purely physical system with computing

power, memory and a communication interface. In the context of Industry 4.0, this combination enables the creation of smart products, production tools and machines as part of the production process. CPS are expected to generate very large amounts of data, which in turn requires the on-demand availability of decentralized, high computing capacity and methods for processing and analyzing the information.

Several CPSs are the building blocks of the concept of Cyber-Physical Production Systems (CPPS). These are production systems that use CPSs and newly defined interfaces between machines and between humans and machines to accomplish the production task. CPPS are designed to enable decentralized, semi-autonomous, cross-enterprise production control (cf. [437], [617]).

Another concept related to Industry 4.0 is *end-to-end digital engineering*. It encompasses the digital mapping of the entire physical production process in a company, from product development to product completion. All planning, control and monitoring processes are constructed and simulated in a virtual environment. The basis is a digital image of the factory with all its physical objects. This includes production facilities, personnel, products and other working and operating resources that are present in the real factory (cf. [617]).

Maturity Levels

The idea of Industry 4.0 is often reduced to a visionary description that represents an already fully realized implementation of all concepts. Although almost all the technologies that would be necessary for successful implementation are already available, it is often only the right combination of approaches and solutions that brings positive results for a company.

The "Acatech Industrie 4.0 Maturity Index" describes a maturity-based development path towards a fully developed Industry 4.0 capability (cf. [584]):

- The first two development levels are not seen as part of Industry 4.0 per se, but form the basis for all further levels. *Computerization* refers to the isolated use of information technologies, while *connectivity* refers to the networking of these isolated systems. Both terms cover the first and second levels of the maturity model.
- The third level describes the ability to *perceive* through extensive equipment of physical objects with sensor technology. This enables the comprehensive collection of data points about interrelated processes. The aim of this stage is to generate the ability to create an up-to-date digital model of reality at any time.
- The fourth development stage describes a state of *transparency* in which, building on the third development stage, interrelationships and causes become comprehensible through the analysis and interpretation of these interrelationships.
- The fifth level describes the presence of *forecasting capability*. For this purpose, the system is equipped with the ability to simulate dynamic processes. On this basis, automated simulation experiments can be carried out that depict different

future scenarios. The evaluation of the experimental results leads to the forecasting capability of the system.
– At the sixth level, *adaptability*, the forecasts are used to derive decisions independently and implement suitable measures automatically.

These six developmental levels represent a development path for the Industry 4.0 capability of a company (see Figure 3.1). In addition, the development levels provide a reference point for classifying existing Industry 4.0 technologies (cf. [584]).

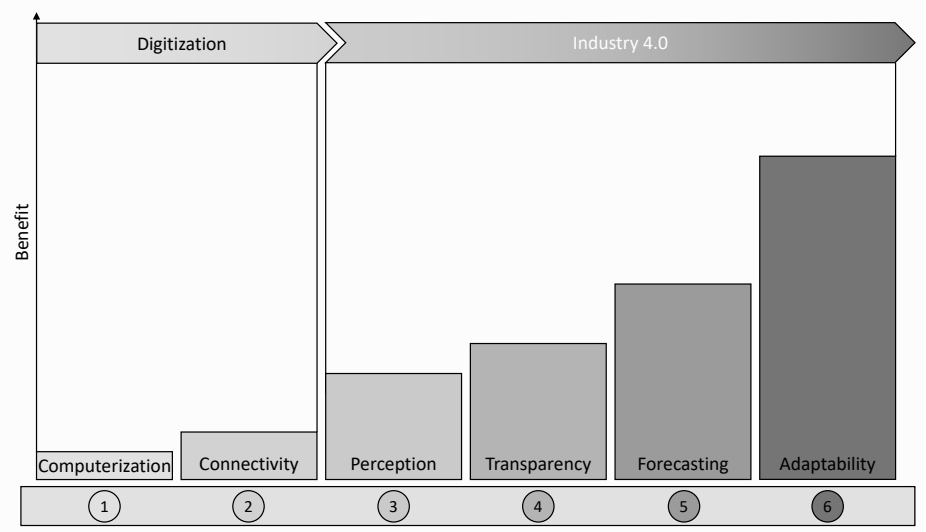


Fig. 3.1: Maturity Index for Industry 4.0.

3.1.3 The Role of AI in Industry 4.0

From the perspective of Industry 4.0, AI technologies can be used to enable a company for reaching the higher levels of the maturity index. AI technology is intended to enable technical systems to autonomously perceive their environment based relying on the data available to them. In particular, it can be used to locate and identify physical objects and determine their state. The subsequent interpretation of the perceived objects and the meaning of their state as well as the relationships between the objects shall help to achieve the necessary transparency for reasoning about industrial process chains. The automatic creation and updating of simulation models for an automated continuous prediction system, as envisioned for the fifth maturity level, could be supported using certain AI methods such as imitation learning.

While AI is sometimes understood in its function as a technological form of human decision-making capability, in the context of Industry 4.0 it is not necessarily intended to directly copy human behavior. On the one hand, AI is intended to achieve the classic goal of automating industrial processes: cost reduction, time savings, quality assurance, etc. On the other hand, AI technologies are expected to manage the emerging complexity of the vast amounts of data coming in from the newly deployed Industry 4.0 sensor systems. To some extent, AI systems may solve problems, discover unexpected issues, or reduce complex situations by uncovering associations in the data hidden to the human eye.

The Data Crisis

However, AI technologies are not "magic fairy dust", but are limited by the underlying data and mathematical models used to try to solve a specific problem [377].

One of the most important engineering issues for the successful use of AI technology in Industry 4.0, especially machine learning methods, is therefore the management of the underlying data collection and its representation in a database. Generating an error-free and uninterrupted data stream is a prerequisite for successful operation. If the data is inconsistent and of poor quality, the digital images of physical objects cannot represent the true physical state. Predictions and decisions made on the basis of such a virtual model would then also be of poor quality.

The design of any Industry 4.0 data stream begins with developing a fundamental, ontological notion about the true nature of the physical object in question. Critical factors are the physical attributes of the object and the selection of sensors as well as the programming of the transformation of sensor data into semantically correct observations. The overarching challenge is that in many places the tools to adequately assess these aspects are still lacking. In particular, in many cases 80 % of the time building a system is spent on data collection and cleaning, a situation that has been called the *data crisis* [377]. "The main reason for the data crisis is the increasing interconnectedness of computers. Access to data is therefore easy and cheap, but its quality is often poor. What we need are cheap data streams of high quality. This means that efficient methods for improving and verifying data quality must be developed."[377] This crisis becomes even more apparent when considering continuous data acquisition by a multitude of sensors, as envisaged for Industry 4.0 systems. Challenges include monitoring the quality of the data stream throughout the life cycle of an Industry 4.0 system and the ability to respond appropriately to changes in the underlying assumptions made at the beginning of the design process.

3.1.4 From Digital to Cyber-Physical Twin

In response to the data crisis, the development of a system should more closely link the design of data acquisition with the design of the physical objects concerned. This results in the concept of the digital twin , which combines the connectivity of a CPS with the virtual representation of objects in end-to-end digital engineering. The concept of the digital twin focuses in particular on the quality of data transmission between the physical and the virtual counterpart.

Like the Industry 4.0 maturity index for companies, the literature on digital twins knows different implementation levels for the digitization of physical objects [354]. The development stages are differentiated according to the degree of automation:

- A digital model has a purely manual data transmission. It can be, for example, a simulation model of a planned factory, a mathematical model of a new product or another model of a physical object that is stored digitally.
- A digital shadow automates the data flow between the physical and the virtual object. Any change in the state of the physical object is transferred and applied to the state of the digital object.
- The digital twin eliminates all manual data transfers. It extends the concept of digital shadow by automating the flow of data directed to the physical object.

In order to be able to map data automatically onto the virtual object, correct identification of the physical object is a prerequisite for both the digital shadow and the digital twin. In addition, the complete elimination of manual data transmission makes the design of data reception at the physical object particularly relevant. Necessary conditions for the digital twin can be described as follows:

- The data of the virtual object must be able to be received and processed automatically at the physical object.
- A physical object can only be part of a digital twin if it can be perceived separately from other objects.
- A perceived physical object must be uniquely identifiable so that the captured sensor data can be assigned to the correct virtual object.

The concrete technical implementation of the identification can vary. It can be done by observation, e.g. by a scanner reading 2D barcodes. It can also be done by communication, e.g. via radio transmission, where recognition and identification result from the defined standards of radio transmission, i.e. how receiver and transmitter can identify each other is defined by communication protocols.

Many of today's Industry 4.0 applications can be classified under the concept of the digital shadow. One example is a low-cost tracker that is built into the bottom of a pallet and records data about the pallet's current location, movement, impacts and temperature profile. This data is sent over the cellular network to a server where the corresponding digital shadow is located (cf. [354]).

Cyber-Physical Twin

Due to the very general metaphor, digital twins can vary greatly in size and complexity. Therefore, the application areas and the possible types of digital twins can be very broad. In particular, there is no distinction between individual physical objects and large, complex systems. The term digital twin can describe a system that includes only a small object or an entire factory. In the case of a large system such as a factory, a central, monolithic approach in the form of a central database would reach its obvious technical limits.

For use in an Industry 4.0 environment, a special class of digital twins is therefore required that provides for a decentralized, modular architecture based on individual physical objects. Their networking should enable scalable, adaptable systems that support autonomous behavior. This new extended concept can be called a cyber-physical twin . It has the following additional characteristics:

- The physical object is considered to be a self-contained and relatively small entity. Its physical extent can be confined to a finite space, and it has a definite location. It is fundamentally mobile, even if it remains in a particular place for a long time.
- The transmission path to the physical object is primarily for changing its behavior rather than issuing direct commands. All behavioral changes are first made to the digital object, with the physical object adjusting its internal behavior accordingly. The behavioral changes can be made by transmitting parameters, compiled source code, or other behavioral data (e.g., a trained neural network).
- The physical object acts largely autonomously and makes local decisions where possible.
- The physical object monitors its environment and can trigger an update process in the virtual image when situations arise in the physical world that require it to adjust its behavior.
- Cyber-physical twins are designed as multi-agent systems that communicate in both the physical and virtual worlds.

Figure 3.2 shows the concept of two cyber-physical twins negotiating with each other in the physical and virtual environments. Note that communication between the environments takes place via an exclusive link between the physical and virtual objects to ensure synchronization of the twins.

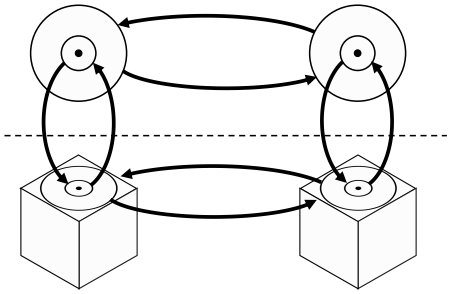


Fig. 3.2: Cyber-physical twin concept

Although not all cyber-physical twins are controlled by AI, the challenges for software and hardware engineering are similar, as high-quality data streams are required in both cases.

The mobility of the physical object is required for the adaptability of an Industry 4.0 system, as it can then change the arrangement of its components. It follows that the embedded computer system of the physical object is in principle resource-constrained, while the computer system in which the virtual object runs is considered unconstrained.

3.1.5 An Excursion into Logistics

Since the introduction of Industry 4.0, the area of *logistics* has been considered its outstanding application domain. In no other area of industry is such a fundamental change expected in the near future. On the one hand, many of the central technical and social challenges are directly or indirectly linked to logistics and efficient supply chain management. On the other hand, this is due to the rapid development of Industry 4.0 technologies. In addition to global data processing capabilities, resource-efficient sensor hardware, communication technologies, and embedded systems are increasingly usable on a large industrial scale. This enables widespread rollout at the level of individual logistics objects (cf. [164]).

In the context of an increasingly volatile production and trade environment, the topology of logistics networks and thus the location of an individual logistics node, such as a transshipment point or a distribution center, can no longer be permanently fixed. In fact, the idea of a fixed, ideal location has not been viable for many years. A logistics network and its nodes must constantly adapt to new circumstances. Therefore, logistics centers should be able to relocate on their own in the future. This rules out many forms of traditional, technical infrastructures and underscores the need to introduce new concepts such as the cyber-physical twin (cf. [279]).

To an increasing extent, swarms of autonomous vehicles will take over intra-company transport. In production systems, the arrangement of workstations can thus

be changed at any time. The vehicles' virtual software agents negotiate orders and business processes, while the physical software agents negotiate movement paths and constantly exchange their locations and those of new stations or storage locations. Autonomous vehicles are able to approach the racks and move bins or pallets in and out. There is virtually no stationary conveyor system in this vision.

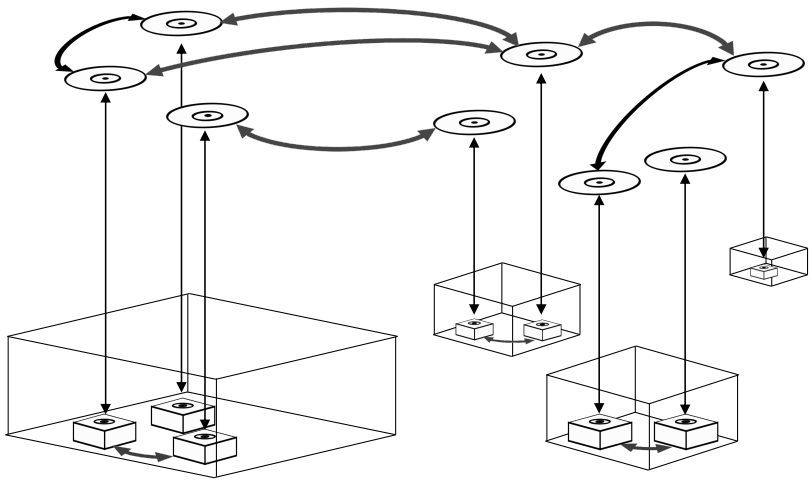


Fig. 3.3: Cyber-physical twins managing warehousing tasks in a supply-chain

Even the shelf and each bin within it can become part of a cyber-physical twin. The bins in the warehouse handle inventory management, check minimum stock levels, and order replenishment (see Figure 3.3). They communicate with shelf displays and transport vehicles. The classic, RFID-based "Internet of Things," as it was devised at the turn of the millennium, is literally getting eyes, ears, arms, and legs.

Direct challenges quickly arise from such all-encompassing networking of physical and virtual objects. Although the absolute amount of data has been increasing significantly in all areas of the economy for a long time, the massive increase in the collection and processing of logistics process data is becoming one of the central challenges to be mastered. The potential for AI applications in cyber-physical twins handling complex logistics processes is significant. From predicting arrival times in transportation and production logistics to dispatching logistics networks and the coming high-frequency logistics, distributed AI solutions can automate and autonomize complex processes that were previously closed to classic control methods. It is expected that AIs will be able to learn how to cope with the complexity of a large number of cyber-physical twins representing logistics objects.

3.1.6 Resource-Constrained Smart Objects That Can Move

The contributions in this chapter address four key areas associated with physical objects in Industry 4.0 systems: the creation of things, their localization, the resource constraints of objects once they become mobile and smart, and their behavior with respect to each other.

The first three contributions deal with the genesis of a thing long before it becomes intelligent or autonomous. Using the example of steel production, machine learning methods are used to investigate how the act of physical separation creates a self-contained entity from a primordial mass and how this separated thing can be assigned to a concrete class by determining or predicting certain properties. An early prediction of undesirable developments enables a controlling intervention in the production process (Section 3.2). The second contribution is dedicated to challenges for machine learning techniques that arise in the context of changing, physical aggregate states (Section 3.3). Combining expert knowledge and collected data in simulation is the topic of the third contribution (Section 3.4).

The fourth contribution deals with the automated localization of things–a basic prerequisite for a continuously (self-)perceiving infrastructure that supports flexible (self-)control of processes. Industry 4.0 thus requires a comprehensive ability to observe physical object movements. Wireless ultra-wideband localization is an example of the technical solution to this task in indoor environments (Section 3.5). In this case, the radio medium is the limiting resource that affects the accuracy and scalability of the application. The contribution is related both to the *smart city* concept (see Chapter 4), which deals with a cross-location infrastructure based on future wireless technologies, and to communication networks (see Chapter 5).

The fifth contribution is motivated by smart things that can move. Free mobility is at its core a deeply logistic property ("the ideal logistic space is empty"). It is also the main cause of resource constraints in terms of energy supply, computational capability and communication for embedded systems. The topic of "Indoor Photovoltaic Energy Harvesting" (Section 3.6) deals in depth with the requirements for ultra-low power devices and their modeling. An important application of this technology is the standardized logistics container, which becomes smart by an embedded ultra-low power computer system. Such smart containers are destined to be one of the key building blocks of Industry 4.0: they are the external interface for the non-smart things that they contain.

The last contribution deals with the behavior of smart and in principle autonomously moving things. It is based on the development of a micro-UAV drone swarm (Section 3.7). The resource constraints of the small drones place high demands on the system architecture. The contribution describes the setup of a testbed environment in which simulation and the physical world are tightly coupled. The drones negotiate their individual movements through simple rules that mimic natural swarm behavior. It is shown how individual rules can be replaced by knowledge learned in

a simulation. The drone swarm testbed illustrates the challenges of developing autonomous swarm systems for Industry 4.0. Since these systems will increasingly be used across companies in business-critical areas in the future, the issue of data protection in multi-agent systems will also become important in this context (cf. Section 6.1).

3.2 Quality Assurance in Interlinked Manufacturing Processes

Jochen Deuse
Katharina Morik
Amal Saadallah
Jan Büscher
Thorben Panusch

Abstract: Interlinked manufacturing processes are characterized by the dependence of downstream process steps on the previous ones. If it can be predicted that a particular workpiece will not reach the desired quality, anticipatory measures can be taken early in the process. By its prediction, machine learning saves resources, both in processing and in the material. One example is the model-based quality prediction in electronics manufacturing on a Surface Mount Technology (SMT) production line. Here, the application of a learned classifier predicting the quality must be fast so that, say, the routing of a piece may be changed. Hence, machine learning itself needs to save its resources, in this case runtime. Another example is the hot rolling mill process for steel bars production. There, several sensors and process parameters deliver process data that need to be aligned and useful features are to be extracted from the resulting stream automatically, in real time. Here, machine learning saves testing time in the factory's quality inspection process.

3.2.1 Introduction

Undetected quality deviations passing through the entire manufacturing chain have a severe impact on internal failure costs due to the increasing rejection and reworking of defective products without being labeled as defective. Therefore, early quality prediction of a specific workpiece indicates whether it will reach the required quality requirements or if some anticipatory measures should be executed in a timely manner to save resources (e.g. time, material) resulting in rejection or further processing. However, due to technological and temporal restrictions, physical product quality inspections are limited to the final process step. In this context, data mining and machine learning techniques can be used to predict the intermediate product's quality, thus gaining transparency on quality properties of intermediate process steps and enabling real-time process adaptation to sustainably increase its efficiency [339, 654].

In general, process modeling can be done on three levels [654]: process understanding; designing better processes and equipment; and online process control in real time. Online control requires integrating data from different sensors at different steps in the process, taking into account the communication between sensors and even

integrating different models in real time. Over the last decades, the majority of works were focused around the first two levels, where models based on domain knowledge have been developed. More recently, with the emergence of the "industrial Internet of Things" or "Industry 4.0" (see Section 3.1), the adoption of different monitoring technologies using sensor technologies [155, 156], has offered new opportunities for applying data-driven approaches for process modeling, in general, and for intermediate quality prediction, in particular.

This section is about making Intermediate Quality Prediction (IQP) along a process chain by embedding data analysis directly into the manufacturing chain. We start by showing how IQP using data mining and machine learning can be integrated into a comprehensive Intelligent Manufacturing Process Control (IMPC) framework for industrial applications. Section 3.2.3 dives deeply into the steps of data analysis. These steps include a detailed description of the data acquisition process, cleansing, the choice of data representation, extracting the right features, , modeling, and evaluation. A framework for processing data streams with the right level of abstraction in real-time, in addition to the real-time management of many machine learning models, is also presented. It glues diverse contributions of data mining and modeling together to form an application.

We present two real-world case studies in Section 3.2.4, starting with the description of the production process and data acquisition followed by modeling and deployment. The first case study addresses the use of data mining in an electronics production environment for the purpose of reducing the quality inspection volume. The case study is conducted on a Surface Mount Technology (SMT) manufacturing line in the Siemens plant in Amberg. The motivation is to relieve the optical end-of-line test, consisting of an X-ray inspection system. The second case study consists of a hot rolling mill process. It showcases the importance of embedded data analytics. The system is developed in close collaboration with experts in machining, production, and the steel mill. Data analysis results are validated by domain experts. The conclusion summarizes our findings and gives an outlook for future research work.

3.2.2 Intermediate Quality Prediction in Intelligent Manufacturing Process Control

Different quality-related tasks for the application of data mining and machine learning in manufacturing can be distinguished [335]: description of product/process quality; modeling of the product quality; quality prediction; and parameters/process optimization.

In literature, there are several standardized procedures for the implementation of data mining or machine learning in general and one approach for optimizing quality-related tasks in particular. Four widely used process models include the Knowledge Discovery in Databases (KDD), the Cross-Industry Standard Process for Data Mining (CRISP-DM), the 5-step SEMMA model by SAS, and the 5A model by SPSS [121, 201, 484,

668]. The general framework for continuously monitoring and optimizing a process under quality considerations is known as IMPC [339]. A characteristic feature of the models mentioned is that they are very similar in terms of their basic objectives and structure. The procedures are subdivided into distinct iterative phases that differ in terms of the number and the content of the respective phases. While the IMPC solely focuses on the technical perspective of a data mining project, some other models include a phase for elaborating the business cases, which indicates the necessary involvement of expert knowledge in every data science project.

Product or process quality description is usually the first step to be performed, especially in the context of highly complex manufacturing systems with non-linear interactions between the different process steps [579]. This model can be based on a physical model designed by domain experts, or it can be a data-driven model using machine learning techniques. The model is subsequently used to predict the quality of new unseen products. Following these predictions, a variety of measures for process optimization can be applied [405, 579, 654]. These measures include early control interventions, optimization of process parameters setting, stabilization of processes, dynamization of inspection plans, and the design of model-based inspection processes.

The IMPC introduces data mining techniques to perform IQP and adjust further production processing steps according to the results of the IQP. The IMPC consists of different functional modules that can be summarized as follows [339]:

1. Data acquisition and storage
2. Intermediate Quality Prediction-IQP
3. Process optimization

While the first module seems to be a prerequisite for building the IMPC, modules 2–3 represent the different process control stages.

The IMPC itself can be realized following two different paradigms. The first one is based on an online optimization of the process based on the current observed state of the process in order to compensate for previous process deviations. Such a decision is made following the domain knowledge provided by production experts and engineers. The second one is a data-driven approach where the first and second modules are integrated into the decision support system. The second type of process control relies on estimating the quality of the intermediate product in real time. Anticipatory measures are taken to prevent possible predicted process deviations. The modular design of the entire process control approach is detailed in Figure 3.4.

The IMPC concept can be viewed as a separate building block in a company's process control landscape [339]. The IMPC does not introduce any change in the process or production structure. However, it generates recommendations in process planning and optimization. Therefore, the focus is brought on analyzing processing states in real time and forecasting the intermediate product's quality properties. This knowledge is used for either deriving recommendations on whether the product should be processed any further or for stopping the processing of the product early, since the final quality

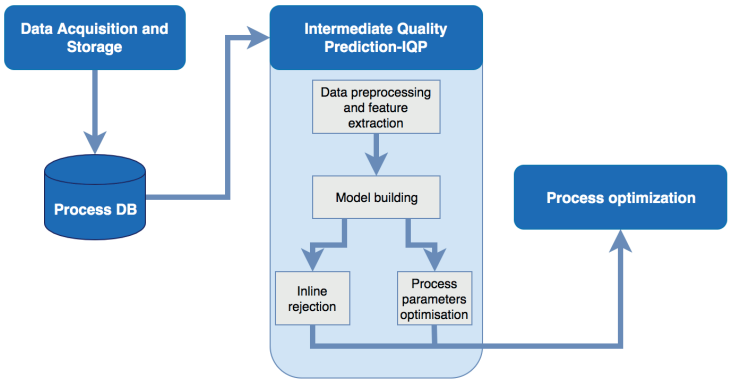


Fig. 3.4: Intelligent manufacturing process control model.

requirements will not be met, or for optimizing the next processing step's parameters to align the product with the required quality standards.

The IMPC requires information on the current processing state and parameters that are gained from the first module, in addition to information on historical processing data. Data is most often collected from sensors implanted to monitor some process variables. Collected data has to be preprocessed and meaningful features should be extracted to be fed afterwards into a quality prediction model in order to assess the intermediate product's quality correctly. Hence, the Intermediate Quality Prediction IQP module translates all available information irrespective of its actual meaning into a quality assessment. This is done by means of data mining, i.e., supervised learning models [339, 654]. The result of the IQP module can then be applied to decision rules or recommendations on process optimization. The process optimization module is the final step in implementing the IMPC model. As described above, it aims at adjusting the parameters of upcoming process steps in such a way that quality deviations caused by previous processing steps are compensated in the remainder of the process chain. It is worth mentioning that when it comes to decision rules following the results of the prediction of intermediate products' quality, even more knowledge is required, including domain knowledge and artificial test-beds (e.g. process simulations) to validate the *correctness* and *feasibility* of the process optimization recommendations.

In the following sections, we will discuss in detail available methods for data analysis and quality modeling using machine learning.

3.2.3 Methods for Data Mining from Sensor Data

The dynamic evolving nature of manufacturing processes poses multiple challenges for data mining from sensor data [654]. For instance, different scenarios may occur, sensors and machines may fail, some deviations in processing steps could be observed

and products may take different routes through the process chain. With such events, difficult challenges on how data should be collected, stored, and preprocessed arise. Additionally, adaptive extraction and selection of which features may be relevant for the prediction task, are necessary. Moreover, to generate intermediate products' quality predictions in real time, stream data mining methods and online management of many machine learning models should be established. The following sections will discuss such problems and available methods in more detail. The structure of the section is depicted in Figure 3.5. It is based on the IQP process steps and comprises standardized procedures that are common steps in every quality-related data-mining task.

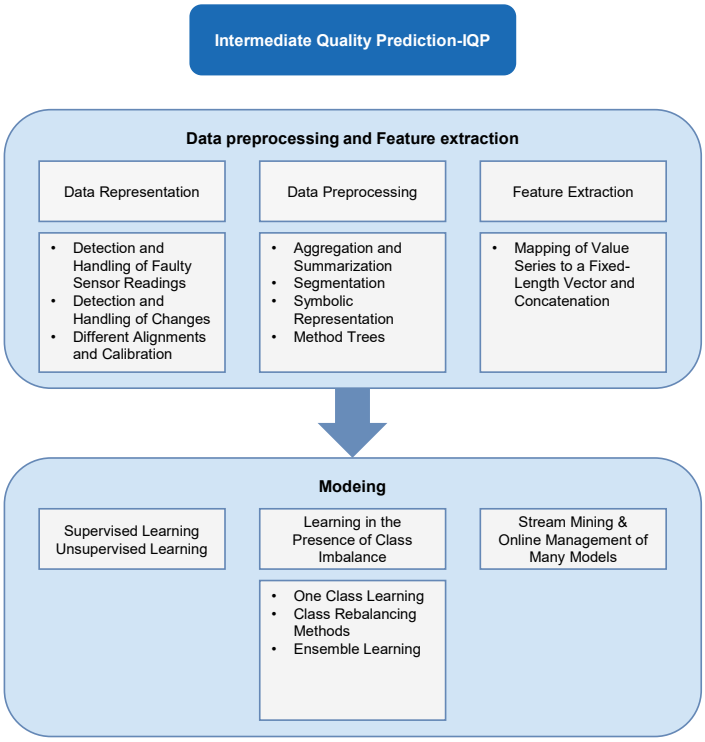


Fig. 3.5: Specification of the IQP process with regard to common steps in quality-related data mining tasks.

3.2.3.1 Data Acquisition and Storage

In intelligent manufacturing systems, data is most often collected using various sensing technologies [339, 405, 654]. The continuous measurements of sensors form an indexed stream of countably infinite data items x_i. Every data item contains an index, e.g., a timestamp, and can contain an arbitrary number of values and value types such as

images, strings, or numbers. A segment of this stream with length n and only numerical values forms a value series that can be defined [445] as a mapping $x : \mathbb{N} \to \mathbb{R}\times\mathbb{C}^m; m \in \mathbb{N}$, where each element x_i of a value series with length n is an ordered pair $(d_i, w_i).d_i \in \mathbb{N}$ is called the index component and $w_i \in \mathbb{R} \times \mathbb{C}^m$ the value component. It is called a time series if the index dimension represents a temporal order. The value component is written using a complex number value space instead of a real number value space in order to formalize sensor readings and their transformations in the same form. In classification tasks (e.g. quality prediction) the goal of a machine learning model is to learn a functional mapping $f(x) : \mathbb{R} \to y : \mathbb{N}^n$, where y formally denotes the class label of the process measurements. In general, this task is named a multi-class classification task. The special case of binary classification occurs when the class label consists of only two complementary classes, say, the Not Ok (NOk) class and the Ok class.

3.2.3.2 Data Preprocessing

In manufacturing environments, sensors may deliver wrong readings or might experience failure periods [474]. Their readings are most of the time noisy. Henceforth, collected data may contain irrelevant readings, be wrongly aligned, or have different resolutions. In an offline setting, such cases can simply be excluded from the analysis or corrected. By contrast, the embedded real-time analysis of data must somehow detect such cases in an automated fashion and react accordingly. The first analysis step therefore usually consists of cleaning the sensor data.

Detection and Handling of Faulty Sensor Readings Faulty sensor readings such as sensor readings lying outside physically meaningful ranges need to be detected. Nevertheless, faulty readings may overlap with the normal data, requiring the automatic detection of faulty patterns using supervised learning techniques (see Section 3.2.3.5) [654]. However, if the faults are not highly frequent, it is difficult to detect them based on available training examples. Models for anomaly detection can be of great help in this context by describing only the normal data. They mark patterns deviating largely from the distribution as anomalies. Nevertheless, the correct definition of parameters, like threshold values, remains difficult with only a few noisy examples. Furthermore, it can be difficult even for domain experts to identify such examples correctly. There are several possible ways on how to handle missing values after being detected including replacement by their predecessor values or auto-regressive moving average approaches [98] or imputation based on predictive models that are trained on other existing values. Additionally, the production process itself might introduce some level of noise to sensor data. If the underlying noise model is known, it should be used. Otherwise, measurements should be filtered and smoothed [405, 654].

Detection and Handling of Changes Deviations and changes in sensor readings may also be explained by intended changes in the underlying hardware, such as when

new production equipment and new or different calibrations of sensors are introduced. Based on the type of changes, it must be decided which of the trained models need to be updated and how. Without any models describing such changes, methods for concept drift detection can be applied and it has to be decided if already trained prediction models must be updated and how these models should be managed [564]. More details are provided in Section 3.2.3.10. Incremental training methods, like streaming methods (see Section 3.2.3.9), can be used to integrate such changes into the trained models automatically.

Different Alignments and Calibration Different data alignments can be observed due to different data sampling strategies and sampling frequencies from sensor recordings. The measurement frequency of a data source is defined by the number of times the source delivers data per time unit. The higher the desired sampling frequency, the more memory it uses. The association and integration of data from many sensors require their synchronization with the environment description. The environment description is characterized by the adequate time of measurement. To obtain a precise synchronization, a sufficiently accurate global time of measurement for the different sensors is required to be defined or derived using synchronization techniques [309]. In this context, a multi-sensors data fusion system has to cope with different and varying measurement frequencies, measurement latencies, and asynchronous measurement times.

Synchronization techniques can be divided into two main families: deterministic and non-deterministic. In a deterministic setting, the measurement times of each sensor have to be known in advance and synchronization is performed on the slowest sensor using aggregation techniques [309]. In a non-deterministic setting, sensors are assumed to be asynchronous and there is no knowledge about measurement times or latencies. In the process recordings, different frequencies might be used. In such situations, recursive filtering approaches such as the Kalman filter or recursive autoregressive filters can be used for synchronization [291].

In addition, different hypotheses for drawing data samples from sensors may also lead to a mismatch in the sense of learning from different underlying distributions. Therefore, sensor data has to be calibrated (e.g. adjustment of sensor parameters, features, raw data cuts for out-ranging measurements using classification or clustering techniques) [654].

3.2.3.3 Data Representation

As it will be shown in the case of steel production (Section 3.2.4.2), a single run through the process chain can be represented by a set of time series with different lengths and offsets, which may overlap in time, contain different numbers of segments at different levels of granularity, stem from different machines, and sensors and may also be entirely missing for optional processing steps.

Many common data analysis methods cannot work directly on such sets of time series; rather, they require that all observations to be represented by fixed-length feature vectors. Here, we discuss how the raw series values can be transformed to fixed-Length vectors and concatenated into one single representation.

Mapping of Value Series to a Fixed-Length Vector and Concatenation The goal is to transform raw sensor data into fixed-length vectors in order to make standard learning techniques applicable. We start by reserving enough space for the records of each sensor in a single fixed-length numerical vector. Original series values are then indexed by predefined positions in this vector. For the mapping, the maximum length of the time series needs to be known beforehand. For the projection, the original time series values might need to be rescaled, e.g., by interpolation. As a result, the application of the most popular distance-based data analysis algorithm becomes possible.

A difficult question is which values to assign to portions where no processing took place. We may simply fill missing portions with zeros or the last recorded value. However, filling with zero values can lead to inaccurate evaluation with several popular distance measures, including Euclidean distance. For example, both series would be marked as highly dissimilar by Euclidean distance, although both blocks (A and B) can have a similar quality. In such a case, the correct mapping between similar feature vectors and similar labels would be altered. Reserving the same portion for both finishing rolls (sensors 5 and 6) in the fixed-length vector seems to solve the problem, but it does not take into account that both finishing rolls might have different properties, e.g., value scales, which usually require a careful data calibration [654]. Instead of transforming all series values to a fixed-length vector, another option would be to use distance measures that can handle value series with different lengths, such as the Dynamic Time Warping (DTW) [461] or the Longest Common Subsequence (LCSS) distance [160]. In principle, two main approaches, for transforming the original time series appropriately, exist. The first approach simply concatenates all time series belonging to the processing of single manufactured pieces (e.g. a single steel block). The second approach consists of calculating distance values for each time series of each sensor independently and then summing them up to a total distance.

3.2.3.4 Feature Extraction and Selection

Instead of using a raw data stream, we can characterize production processes by a devised set of features extracted from raw data. The transformation of the raw data into a feature vector should maximize the prediction accuracy and increases the resource awareness of the machine learning algorithms by reducing the dimensionality and the amount of stored data. (See Chapter 1 in Volume 1.) The challenge in this task consists of the huge search space (i.e. exponential size) over all the possible transformations. Traditional approaches are based on manual feature extraction to build features one at a time using data analysis and domain knowledge. While these approaches are

accessible and interpretable to users, they are tedious, time-consuming, error-prone, and usually not adaptive to data changes. In this context, automatic feature engineering appears to be a promising way to go. It consists of automatically extracting and generating a large number of features and selecting an effective subset of these features to ensure better performance of the machine learning algorithm. Many works have been conducted for automated feature engineering in literature either by automating the whole process [594],[445], or by focusing on particular steps such as feature extraction, feature generation [314] or feature selection [582].

The following sections present a non-exhaustive list of transformation and feature extraction methods that look especially promising in the context of production processes.

Aggregation and Summarization Aggregation and summarization methods for streaming time series data reduce the amount of collected data as much as possible and try to retain its most important patterns. The simplest type of aggregation is based on the calculation of summary statistics such as minimum and maximum values, the mean, median, standard deviation, percentiles, and histograms. Such simple features can already encode sufficient information for the learner and sometimes outperform sophisticated methods [654]. More sophisticated methods search other representations of the time series based on time series transformations. These include the Discrete Fourier Transform (DFT) [445] and the Discrete Wavelet Transformation (DWT) [445].

Segmentation The salient features approach by Candan, Rossini, Sapino, and Wang [115] transfers ideas from the segmentation of two-dimensional images and the extraction of Scale Invariant Feature Transformation features (SIFT) from images to the space of one-dimensional value series. Salient points in the series, which are points that deviate much from their surrounding values, are used for segmenting the series. Then, from each segment, characterizing features are extracted. The method determines salient points at different resolutions, allowing for a description of value series at different levels of granularity. One segmentation approach was developed in the context of the steel production use case (Section 3.2.4.2), which determines segments based on domain knowledge and signals from machines in the process chain [654]. After segmentation, different statistics are computed on the segments such as the mean, the standard deviation, and the minimum and maximum values. Other features are differences between values and histograms. The biggest advantage is that the approach allows for the combination of many features in a highly interpretable manner [654], since the feature transformation is handled in a multivariate manner in the sense that features from different value series and their parts are combined together. For example, a classification rule that is formed based on such features may be: “Predict a steel block as defective if it was heated for less than one hour at 900 degrees and if the maximum rolling force in the first rolling step exceeds the value of 10 000”. The approach has

already been used successfully for the identification of coarse-grained patterns such as processing modes (see Section 3.2.4.2).

Symbolic Representation SAX (Symbolic Aggregate Approximation) [413] first determines the elements of a sequence $C = (c_1; \cdots ; c_n)$ by piece-wise aggregate approximation and maps them to a new sequence $C^{'}$ with $w < n$:

$$c_i^{'} = \frac{n}{w} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} c_j \tag{3.1}$$

The elements $c_i^{'}$ are then discretized by mapping them to a fixed number of symbols, keeping the upper bounded Euclidean distance between all series. A gradient-based approach for the symbolization of streaming sensor data was introduced by Morik and Wessel [458]. This approach was originally applied in the context of text mining and has been successfully used in areas such as text classification or intrusion detection.

Method Trees All of the aforementioned methods, with different parameter settings and combinations, are useful for extracting functional features from value series. Instead of trying and combining all the existing methods and different parameter values manually, Mierswa and Morik [445] developed an automatic representation learning method that optimizes the composition of a representation for best classification learning. Basis transformations, filters, mark-ups, and a generalized windowing present elementary methods that are combined in the form of a method tree. The tree applies the operators (nodes) in a breadth-first manner to transform the original value series. The root of each tree consists of a windowing function, while the children of each parent node are operator chains representing basis transformations, filters, and a finishing function. Learning the feature extraction tree is performed by a genetic programming algorithm. The method can be used for the analysis of time series from production processes. However, its demand for stratified datasets with respect to the labels is not always met by the production data. The method had been implemented within RapidMiner.

3.2.3.5 Modeling

The following sections give a short overview of widely used methods that are assumed to be of special relevance for the embedded data analysis in production processes.

3.2.3.6 Supervised Learning

The most popular task in the context of Supervised Learning is inferring functions representing the relationship between a set of explanatory variables (i.e. features) and a response target variable which can be discrete (i.e. classification) or continuous (i.e. regression), also known as *function learning*. Let X be a set of possible explanatory

variables, Y be a set of possible target values and D an unknown probability distribution on X, Y. Let further H be a set of possible functions. Given examples$(x, y) \in X \times Y$, drawn from D, where $y = f(x)$ for an unknown function f, the goal is to find a function $h \in H : X \to Y$, such that the error $er_D(h, f)$ is minimized.

In the case of a production process, explanatory variables are feature extracted from sensor measurements (i.e. records for a given steel block) and $y \in Y$ is the corresponding quality label to be predicted. Examples of learning functions $h \in H$ include Decision Trees, K-Nearest Neighbour (kNN), Naive Bayes and, Support Vector Machines (SVM). For a detailed in-depth overview of such methods, see Hastie et al. [260].

The aforementioned methods assume that all training data is available in batches. Hence, they cannot be trained during the manufacturing process, and their models require to be retrained if concepts change. In case of the absence of concept drifts, models can be trained offline but deployed online for the prediction.

3.2.3.7 Unsupervised Learning

If no labels are provided, unsupervised learning methods may be employed to reveal the most prominent patterns in the data. Cluster analysis [300] tries to group observations, following a similarity measure. The number of clusters k is usually a user-defined hyper-parameter. A well-known clustering algorithm is k-means [426]. Dimensionality reduction techniques such as principal component analysis (PCA), aim at simplifying high-dimensional datasets. Some of them may be used for better data visualization, like SOMs [654], which map high-dimensional input vectors to a low-dimensional grid. Vectors that are similar to each other in the input space lie close to each other on the grid. SOMs have also been used for analyzing the data in the steel production case study (see Section 3.2.4.2). The biggest disadvantage of unsupervised methods is that, without any labeled data, their results can only be validated by domain experts.

3.2.3.8 Learning in the Presence of Class Imbalance

Class imbalance occurs when data classes are not equally frequent. Generally, it occurs when some classes represent rare events, while the other classes represent the counterpart of these events. Rare events, especially those that may have a negative impact, often require informed and prompt decision-making. However, the class imbalance is known to induce a learning bias towards majority classes, which implies a poor detection of minority classes. For example, production processes with high-quality standards usually output more high-quality goods. Similarly, certain events, like machine or sensor failures, may only occur rarely. In such cases, many positive examples but only a few or even no examples of the negative class are available. Measuring class imbalance performance in classification tasks based on the accuracy leads to the problem that the metric is biased towards the majority class. Class imbalance can be mitigated using different methods including one-class learning, class rebalancing methods, and ensemble learning.

One Class Learning One way of handling class imbalance is to treat minority class instances as outliers or anomalies and majority class instances as the normal class using the task of one class learning [460]. Tax and Duin [675] propose a Support Vector Data Description (SVDD) that computes a spherical boundary around the given data points. The diameter of the enclosing ball and thereby the volume of the training data falling within the ball can be chosen by the user. Observations inside the ball are then classified as normal whereas those outside the ball are treated as outliers or anomalies. Schölkopf et al. [581] have proposed the 1-Class SVM, which separates all training examples with a maximum margin from the origin. An active approach to such data domain descriptions generalizes this approach [238].

Class Rebalancing Methods Several methods have been proposed to handle the class imbalance problem using resampling of the data [563]. Resampling strategies rebalance the data in order to mitigate the effect of the bias of machine learning models towards the majority class [125]. Resampling methods are considered to be flexible and are widely used since they are independent of the selected classifier. The class imbalance problem can also be solved using algorithmic modifications of existing machine learning classifiers, e.g., support vector machines, k-nearest neighbors, or neural networks. Modifications can be introduced by enhancing the discriminatory power of the classifiers towards the minority class using kernel transformation to increase the separability of the original training space [219].

Cost-sensitive learning can also be applied in the context of class imbalance by modifying the loss functions to increase miss-classification costs of the minority samples [322].

Learning Ensembles in the Presence of Class Imbalance Combining classifiers in ensemble frameworks is another common approach to handle the class imbalance problem [563]. Within ensemble-based classifiers, we can distinguish four main families. The first family includes resampling based ensembles. An ensemble of classifiers is created after training base classifiers on balanced datasets obtained with a resampling technique. In the second family, the ensemble is built based on boosting [576] after applying a data resampling strategy (e.g. SMOTEBoost [126]). Within the third family, we find Bagging-based ensembles [101] (e.g. UnderBagging, OverBagging, SMOTEBagging [715]).

Recently, we proposed a probabilistic ensemble method to handle the class imbalance explicitly at training time [563]. Unlike existing ensemble methods for class imbalance, which use either data-driven or randomized approaches for their construction, our method leverages both directions. On the one hand, ensemble members are constructed from randomized subsets of training data. On the other hand, we design different scenarios of class imbalance for the unknown test data. For each of the resulting scenarios, an ensemble is obtained by combining random sampling with an

estimate of the relative importance of specific loss functions. The final predictions are generated by computing a weighted average over the individual ensemble predictions. In contrast to existing methods, this approach does not attempt to correct imbalanced datasets. Instead, it has been shown how imbalanced data sets can facilitate classification, given a limited range of true class frequencies. This method promotes diversity among ensemble members and is insensitive to certain parameter settings.

3.2.3.9 Stream Mining

To cope with real-time evolving production systems, every step of the analysis should be applied and handled in an online fashion. Whereas learning an effective representation needs to be run offline as shown above with model trees, using the learned extraction must be applied online. Many applications demand the transformation of the raw sensor data, the deployment of the model, and sometimes even the model update and the detection of novel events being performed online. Change detection is usually done by updating the model with every new arriving data item or learning a model on a batch of the most recent data items [218].

The biggest challenge in applying classical stream mining settings to production processes is, that the labels are usually measured at the end of the process chain. Hence, at each point in time we predict the final quality. For the time series starting with t_{i0} and ending with tie, we have the observations $x_{ti0}, ...x_{tia}, ...x_{tie}$, but only the final quality measurement y_{tie}. Hence, at each observation point we predict the final quality. The model f uses p extracted features from $x \in X$ (i.e. $f(\phi_1(x), \cdots, \phi_p(x)) = y$). Assume t_{i0} to be the starting time of the process p_i, t_{ie} its end, and t_{ia} the actual time instant, one possible approach is to segment the time series and learn an individual model on every single segment to predict the final quality. This approach is useful when important events can not be identified in time and memory constraints are imposed, i.e., training models on time series subsequences instead of considering all the historical data. If it is possible to identify important events within the corresponding step of the process occurring at a given time instant t_{ic}, then every feature extracted from the segment $[t_{i0}, t_{ic}]$ can be used to learn a model. It is therefore possible to generate the first prediction of the label if $t_{ia} > t_{ic}$.

A second approach consists of using a combination of static and statistical features, like the minimum, maximum, or average of the time series. The static data won't change over the complete process and the probability of change of the statistical features will decrease to the end of the process. That means, that there exists an index t_{is}, where the prediction error is bounded by $\left(\hat{f}(\phi_1([t_{i0}, t_{is}]), \cdots, \phi_p([t_{i0}, t_{is}])) - y\right)^2 \leq \xi$. Another approach would be to use one or a set of algorithms, that predict the set of features for the unseen part of the time series $[t_{ia}, t_{ie}]$ and train a model on the full feature set to predict the quality label. The overall prediction will therefore be strongly dependent on the accuracy level of the feature prediction.

3.2.3.10 Online Management of Many Models

As it is mentioned in stream mining, machine learning models can be trained offline and deployed in real time to generate quality predictions. It may also happen that the stream-mining setting requires the training of many models on different parts of the data, e.g., different segments of the time series (see Section 3.2.3.9). In addition, the increasing individualization of products and processes may result in small heterogeneous groups of observations. Learning distinct models is then necessary to represent each group of observations and capture its underlying properties. Therefore, efficient online management of many models should be established through the dynamic combination of many models built to comply with detected changes in the data by adaptively changing their combination and integration rules in real time. The dynamic combination can formally be established with adaptive ensemble methods that are built as a weighted combination of distributions characterizing the target concepts and enabling flexible management of the models from the individual model selection to the weighted aggregation of many models [561, 562, 564, 565]. The application of adaptive ensemble methods is made in connection with concept changes detection in the data that enable the ensemble update on different levels (i.e. base models selection [561], informed base models/ ensemble parameters adaption (i.e. after a detection of a concept drift) [562]).

In this context, we have proposed an adaptive ensemble selection framework that manages online two main ensemble construction stages: pruning and integration [564]. Since the performance of ensemble-based models changes over time, it is also considered to be subject to concept drifts. A drift detection mechanism is employed to exclude models whose performance becomes significantly worse compared with the remaining models and to identify the top base models in terms of performance. Performance is evaluated in this context using a custom measure based on the Pearson Correlation (i.e. commonly used to deal with time series data between base models forecast and target time series on a sliding window validation set. After each drift detection, top base models are identified. Since diversity is a fundamental component in ensemble methods, we perform a second stage selection through clustering model outputs. Clusters and top base models are updated after each drift detection (i.e. whenever an alarm is triggered by our drift detection). At each cluster computation, the models that belong to the cluster representatives are selected. In a final step, the selected models' outputs are combined together using a sliding-window weighted average.

We have also developed a framework for online ensemble aggregation using deep reinforcement learning for time series forecasting [562, 565]. There, we leverage a deep reinforcement learning framework for learning linearly weighted ensembles as a meta-learning method. In this framework, the combination policy in ensembles is modeled as a sequential decision-making process and an actor-critic model, that aims at learning the optimal weights in a continuous action space, is used. The policy is updated following a drift detection mechanism for tracking performance shifts of the ensemble model over time [562].

These frameworks were developed initially to manage forecasting models and can be applied to predicting continuous quality-related measures over time, but can also be transferred to classification models operating in streaming environments.

3.2.4 Case Studies

The following case studies highlight different facets of the introduced theoretical framework and give a decisive impression of how the described methods can be applied in practice, in order to solve quality-related problems. However, not all concepts have been validated in the particular industrial environment. In the SMT manufacturing use case, the steps of data acquisition and storage, learning in the presence of class imbalance as well as feature extraction and modeling are explained. Lastly, the process optimization in the SMT line is described. In the hot rolling mill process, the data acquisition and storage are highlighted, since the processing of the sensory data was relatively challenging due to incomplete sensory measurements and other factors. Also, the feature extraction and modeling phase are explained.

3.2.4.1 Model-Based Quality Prediction in Electronics Manufacturing

The case study of electronics production covers the production of programmable logic controllers of the Simatic type. At the end of the soldering process, the correct position of soldered components is checked. This is accomplished by an Automatic Optical Inspection (AOI) for variants with visible connection points (pins) and by an X-ray inspection for variants with the pins underneath the components from two different perspectives (X_1 and X_2). The Printed Circuit Boards (PCB) are placed on a panel and are tested in a pool test by placing 48 PCBs in eight Fields Of View (FOV). The number of pins on each PCB are 79 and 52 for X_1 and X_2 directions, respectively. A typical PCB board is illustrated in Figure 3.6 to showcase one typical product variant that is manufactured using the SMT technology. Due to the long inspection time as well as the large number of units produced, X-ray inspection is a bottleneck in the production of PCBs, especially when considering the 100 % inspection, i.e. the testing of every PCB, that is conducted and the constantly growing demand for programmable logic controllers [579].

For this case study, the focus has been narrowed to one product variant, its respective manufacturing line, and the data sources of SPI and X-ray inspection.

Data Acquisition and Storage Historic datasets from Serial Peripheral Interface (SPI) and X-ray are matched by different manufacturing databases via a unique identifier. Resolving the use case at the pin level, where each pin is assigned a quality label, is a low-dimensional machine learning task as was shown by [580] since only seven different features are mapped to one categorical class label. However, from a process perspective,

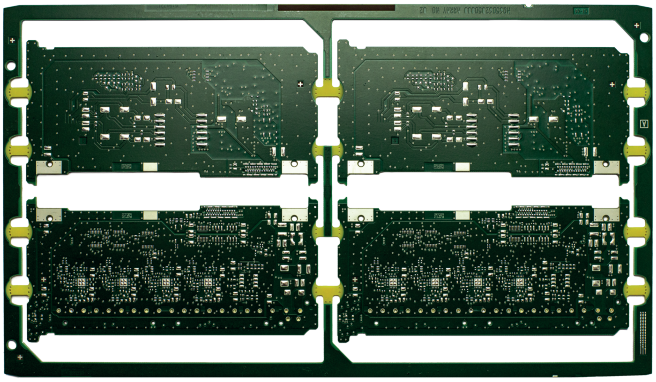


Fig. 3.6: The image illustrates a PCB board from the X_1 direction that is manufactured using SMT.

the quality test is not feasible on a pin level and the process data is aggregated to the FOV level. The dimensionality of the process data thereby increases dramatically. The considered dataset consists of numeric SPI features (see Table 3.1) and a binary X-ray label on the aggregation level of FOVs, which is formalized as a binary classification task. The features characterizing each pin are summarized in Table 3.1.

Tab. 3.1: Descriptive PCB features on the pin level.

SPI feature	Unit
Height	%
Shape 2D	%
Shape 3D	%
Surface	%
Volume	%
Offset X	μm
Offset Y	μm

Historical datasets for a period of five production months are used for the case study. In total, 1 461 037 321 data points are parsed, of which 800 parts per million (ppm) are NOk.

Non-representative datasets, which, for example, are recorded under obsolete process configurations or during manufacturing trials, are eliminated using expert knowledge. The result is a prepared and cleaned training dataset for subsequent modeling, including a unique identifier, all relevant features and the quality label, which can be continuous or discrete depending on the applied measurement method.

Learning in the Presence of Class Imbalance The considered use case constitutes an extraordinary case of class imbalance. When using no specific countermeasures, the classifier degenerates due to frequent misclassifications of the minority class. Classifying the defect class as OK is often unacceptable in an industrial environment. Therefore, specific measurements are chosen in order to increase the performance of the algorithm on the NOk class. To rebalance the dataset, a combination of different resampling techniques are examined, such as oversampling (SMOTE, random oversampling) and undersampling (random undersampling). Furthermore, the output class-membership probability threshold is systematically tuned in order to achieve a better trade-off between both the precision and recall of the classifier.

Feature Extraction and Model Learning Initially, the PCB process data was described on the pin level. Process expertise led to the conclusion that aggregation to the FOV level is necessary so that each FOV consisted of either 79 pins in the X_1 direction, or 52 pins in the X_2 direction with seven features for each pin. While [580] results on the FOV level were not promising, further experiments are conducted. One direction is to reduce the dimensionality using automatic feature extraction methods in combination with machine learning models. For this we use the Rapidminer extension Value Series Plugin. This reduces the dimensionality of the process data for X_1 from 553 features to 240 features and from 364 features to 50 features for X_2.

Model Evaluation The training of the models takes place in a nested structure of inner and outer cross validation and hyper-parameter optimization. As the a priori selection of adequate algorithms is not achievable in a generalized way, different learning methods and algorithms are tested and evaluated for each individual application [579], including Gradient Boosting Trees (GBT), Random Forest (RF), and Multilayer Perceptron (MLP), which is based on fully connected neurons that compute a complete weighted sum in the affine transformation stage of the connections. These methods are used as a baseline for the experiments. Opposed to a fully connected neuron, a 1-Dimensional Convolutional Neural Network (1D-CNN) is based on the principle of weight sharing on all connection units. For example, in the discrete one-dimensional (1D) case: $(d \star K)(x) = \sum_a I(x - a)K(a)$ is a convolution of a data sample d with the kernel (or filter) K [767]. A 1D-CNN with a kernel size of $k = 3$ is used in order to integrate knowledge from the direct neighborhood of pins in question into the learning process. Two convolutions are stacked and, additionally, dropout with a rate of $d = 0.5$ is used for regularization. The most discriminative features are obtained by combining max pooling and two consecutive dense layers. The best performing rebalancing technique on this use case is random oversampling which is applied before feeding the data to the classifiers. The summarized results can be found in Table 3.2.

The model results indicate that the overall performance of the classifiers is modest when it comes to overall correctness. The GBT model shows the overall best performance,

Tab. 3.2: Cross-validated classification results.

Metric	Recall(NOk)	Recall(Ok)	Accuracy
RF	69.83 %	53.76 %	73.46 %
GBT	73.98 %	39.70 %	43.26 %
MLP	16.08 %	**95.13 %**	80.15 %
1D-CNN	**75.4 %**	43.85 %	**74.71 %**

but the difficulty in this use case is the high-class imbalance rate. It appears that the distribution of the NOk class is not accurately represented by the classifiers or only at the cost of a lower recall on the Ok class. However, from a practical perspective, the 1D CNN can be used in order to reduce the testing effort of the X-ray machine, which can save up to 75.4 % of testing volume, which is shown in Table 3.2 as the recall of the NOk class. Surprisingly, the experiments using feature selection methods do not improve the performance as we expected even though the dimensionality is reduced. It appears that in this use case, every individual pin has to be considered, summarizing the data through statistic features leads to an information loss. Lastly, since deep learning models usually lack interpretability, it is not clear which discriminative features led the classifier to draw its conclusion. By contrast, tree-based methods offer more interpretability. Therefore, the decision for a specific model is also a question of its accuracy and to some extent its interpretability by domain experts.

Process Optimization The model deployment is achieved through organizational integration into the inspection planning process. As described by [580], the models were trained on a cloud infrastructure and stored on an edge device close to the process to reduce latency and bandwidth issues. Here, the inspection strategy determines the role of the model with respect to inspection planning and design. While an inspection, exclusively based on the prediction model, requires high confidence in the model and extremely high model accuracy to reach or exceed the level of conventional inspection principles, hybrid approaches seem promising for the current state of development. The inspection reliability is given by the combination of quality prediction and conventional inspection. The introduction of quality prediction in quality assurance can facilitate the generation of additional added value by reducing physical inspection volume without sacrificing inspection reliability. Two different strategies can be deduced depending on the trustworthiness of the model. Either only those parts are subjected to a physical inspection whose prediction result was Ok or whose result was NOk. As the class imbalance of datasets in the quality context is usually quite high, selecting only NOk predicted parts to undergo the physical inspection offers vastly superior potential savings.

3.2.4.2 Real-Time Quality Prediction in a Hot Rolling Mill Process

In the hot rolling mill case study, steel blocks move through a process chain to become bars. The process chain is shown in Figure 3.7. First, the blocks are heated for 15 hours in five different heating zones of a furnace. They are then rolled at the block roll and the first finishing roll. The rolling in the second roll is optional.

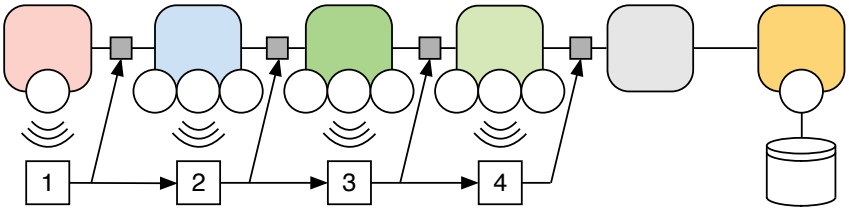


Fig. 3.7: Sensor measurements for two different blocks. ©[2016] Springer. Reprinted, with permission, from [654].

Each block usually moves over a single roll several times, where the number of rolling steps, each taking a few seconds, is determined. The blocks are finally cut into smaller bars whose quality is assessed using an ultrasonic test, several days later. Online measurements on how the blocks are processed are provided by sensors installed along the production chain. The sensors measure various physical qualities including the air temperature, rolling force, rolling speed, and the height of the roll, with 10 Hz. The ultrasonic test results indicate the amount of material containing defects for each bar. It is impossible to assess the physical quality of hot steel blocks or smaller bars at intermediate steps of the process chain. The blocks must cool down before their final quality can be tested in an ultrasonic test. Quality deviations are assessed on specific internal quality parameters such as location, manifestation, and frequency of core and border displays [404]. In cases where some of the blocks are, for example, wrongly heated, energy, material, and human workforce are wasted if they nevertheless continue the process chain. Therefore, the goal of the case study is to identify quality-related patterns in the sensor data, and to predict the final quality of steel blocks in real-time, hence detecting NOk blocks as early as possible. Energy savings can be achieved if, following the quality prediction, blocks with estimated defects are sorted out from the process early enough or reinserted again in previous steps. In addition, it may happen that the required final quality can be reached by adjusting the parameters of subsequent processing stations [339].

In the following, we describe in more detail which sensor measurements are recorded and how they are stored, preprocessed, and analyzed in the context of the given case study. See also the Section 3.3, which presents a novel learning method inspired by exactly this use case.

Data Acquisition and Storage During the period of one year, over one billion measurements from 30 different sensor types are collected during the processing of about 10 000 steel blocks, together with the corresponding quality information. Among the readings are the air temperature for each furnace zone, rolling speed, force, position, and temperature. Domain experts consider these to be the most relevant quality-related parameters. For validation purposes and guaranteeing the reproducibility of results, all data has been stored in a single SQL database. A Java tool was developed for reading and importing the raw data delivered in different files and formats. Once imported, sensor measurements can be exported based on filters written in SQL to CSV files, that contain all measurements recorded by a particular sensor during the processing of a single steel block. For the preprocessing of time series data in production environments, a highly modular process has been developed and implemented with Rapidminer [405, 654]. The following sections provide a summary of the procedure and results already presented in [404, 654]. At first, all value series are cleansed by cutting away irrelevant parts where no processing happened, as discussed in Section 3.2.3.2. In addition, data measurements that lie outside meaningful ranges are marked as outliers and replaced by their predecessor value.

Afterwards, the time series are segmented based on background knowledge as described in Section 3.2.3.3. In the case of the heating furnace, for instance, the five different heating zones make up natural borders for the segments. Similarly, individual rolling steps are natural divisions for all series stemming from the three different rolls.

Feature Extraction and Model Learning Each segment in the time series is described by several statistics, and mapped to portions of a fixed-length vector. The 60 000 raw series values recorded for each steel block are aggregated to about 2000 features. The resulting dataset can then be fed to common feature selection and learning algorithms. For 470 processes, the mapping of the resulting cut bars to the steel blocks could be established. The feature vectors of these processes have then been used for comparing diverse machine learning methods: Naive Bayes, Decision Trees, k-NN, and the SVM [404, 654]. It has been shown that including features about the individual segments decreases accuracy in comparison to including global information about the value series and segments. Features of individual segments were therefore excluded for the following analysis, resulting in 218 remaining features.

Model Evaluation Even with extracted and selected features, none of the classifiers mentioned is able to reach a better prediction accuracy than the baseline, which predicts the majority label. For getting a better impression of the data, the feature vectors were mapped to a two-dimensional Self-Organizing Maps (SOM) (Figure 3.8) where points close to each other have similar features (see Section 3.2.3.4). The shading is used to indicate a weighted distance between the points, where lighter shades represent larger distances. In the SOM on the left-hand side, the points represent the feature vectors of

production processes and their colors indicate the final quality of the resulting steel bars as Ok (red) and NOk (blue). In many cases, NOk bars are very close to Ok ones (see also the zoomed area in Figure 3.8), which means they have highly similar features. As a result, the features extracted so far do not suffice to correctly classify low and high-quality processes.

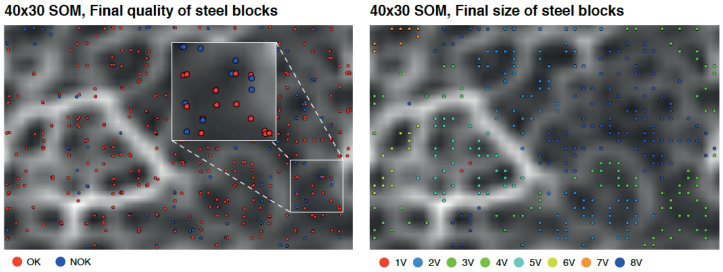


Fig. 3.8: Similarity relationships between feature vectors. ©[2016] Springer. Reprinted, with permission, from [654].

In comparison, the SOM on the right-hand side of Figure 3.8 shows the final size of the produced bars. As it seems, the extracted features are highly correlated with distinct operational production modes for the different bar sizes. The hypothesis could be verified by training a decision tree on features of the first finishing roll. The accuracy as estimated by a 10-fold cross-validation is 90 %, while k-NN ($k = 11$) even achieves 97 %. Most important for the decision is the position of the roll (sensor 501). This indicates the height of the roll and naturally implies the pressure on the block. This correlates with the size of blocks after milling, expressed by $1V$, $2V$, ... Domain experts have verified that the results reflect the real operational model in the rolling mill. That the features are correlated with distinct operational modes and not the quality could mean that large absolute quantitative differences between the modes, i.e. the global patterns, overshadow local patterns. The clustering with SOMs thus gives important hints for improving the quality prediction. For instance, in the future, separate models for the modes could be trained, more scale-invariant features could be extracted, or the value series could be better normalized.

As the results demonstrate, data analysis methods are capable of detecting meaningful patterns in production processes. Even though identifying the exact features that are relevant for the prediction task is sometimes not straightforward, as already discussed in Section 3.2.3.4, each insight into the data can be useful for providing new ideas for better feature extraction and process understanding that can be validated afterward by interaction with domain experts.

3.2.5 Conclusion and Future Outlook

This contribution gave an overview of data mining for Industry 4.0. The first sections offer a general guideline for applications similar to ours. In Section 3.2.2, we present the need for intermediate quality prediction in order to reduce material and energy consumption and how it can be integrated into an IMPC module in order to prepare factories for the adequate use of data mining techniques in a quality-related context. Carefully going through all the steps of the data-mining process in Section 3.2.3, we present a large variety of methods for each step. The contribution presents two real-world case studies in Section 3.2.4: quality prediction in electronics manufacturing (Section 3.2.4.1) and in a hot rolling mill process (Section 3.2.4.2). Both use cases demonstrate the benefits of real-time embedded data analysis in the production chain for quality prediction.

A new application of multiclass time series classification predicts the quality of a bolt in a real-world automotive industry use case. Shorter subsequences of time series were determined that already allowed to train a model achieving high recall and F-measure (both 97 %) for almost all of the 8 classes except for the two that were only present in 1 % or 5 % of the data. Detecting the quality as early as possible enables to do corrective actions, thus avoiding costly rework and waste of resources through further processing of defective components. Moreover, anticipating the type of defect helps to estimate the reworking time to correct it, which varies from 5 seconds to 5 hours [557].

Another more recent study is about saving quality testing efforts in surface mount technology. Since it would take too much time to assess the quality at pin level, the quality information of the panels is aggregated at a FOV level, which corresponds to the aggregation level of the X-ray inspection. One FOV consists of 6 PCB and is denoted as "NOk" if one PCB is detected as defective, whereas it is declared as "Ok" when all PCBs are defect-free. In addition to excellent recall and accuracy, the trained model was explained using a heat map [558]. This is an important step into the interpretability of learned models, but further work is needed.

Despite several success stories, there are still limitations that slow down the application of the developed machine learning-based solutions in the industry. The first of the two most-important obstacles that are still in the way of machine learning adoption in real environments is the lack of reliable labeled data in many manufacturing scenarios. Data gathering, data fusion from heterogeneous sources, and data cleaning require ongoing efforts. The second is that the internal organization of companies needs to integrate computer scientists with a qualification in machine learning into the higher levels of engineering departments. The social integration of employees with diverse backgrounds not only at the board of directors is a challenging task, but it is necessary to benefit from the full potential of machine learning.

3.3 Label Proportion Learning

Marco Stolpe
Katharina Morik

Abstract: For the interlinked process of producing steel bars, quality information is given in statistical form for whole charges of blocks, i.e. we know the fraction of blocks which had quality related problems. This poses a new kind of problem for machine learning, since almost all supervised learning methods assume to be given labels for individual observations instead of groups. The problem of learning from fractions of labels has become known as the problem of learning from label proportions. In this contribution, the learning problem and existing methods for solving it are introduced, as well as a clustering-based method called *Learning from Label Proportions by Clustering* (LLPC) developed in the context of project B3. It is demonstrated that LLPC outperforms methods that were considered the state of the art at that time in terms of prediction and runtime performance. Moreover, the relation to resource-constrained learning settings such as distributed learning is shown.

3.3.1 Introduction

In smart manufacturing, it can be difficult to track products through the whole process chain. For instance, in the interlinked production process of hot rolling, the temperature of steel blocks is so high that they cannot be stamped or equipped with RFID chips. Once cut to smaller rods, tracking object identity, i.e. which rods belonged to which block in which customer order (or batch), can therefore become a big technical challenge. In the steel-rolling scenario, quality labels are usually given as percentages for whole batches, but not for individual blocks. This is similar to other industries, in which due to cost reasons only the quality of a small sample is checked. Depending on the estimated fraction of faulty products, either the whole batch needs to be thrown away, checked again, or it is accepted that a very small fraction of faulty products is delivered to the customer. An interesting question is if we can check only a sample, but nevertheless derive information about the properties of individual products. Moreover, can we sort out some rods already during the production process itself, before the quality check? To put it in more general terms: can we derive a model that assigns correct labels to individual products based on their properties, if we are only given the proportions of each label for different batches of products?

The aforementioned problem of *learning from label proportions* (LLP) not only has applications in industry, but in application areas as diverse as privacy-preserving data mining, election forecasting, bank customer classification, bankruptcy predic-

tion, or marine litter beaching prediction. Especially relevant for us is its relationship to resource-constrained distributed learning settings. Distributed machine learning receives subsets of the overall data in two ways, horizontally and vertically. Horizontally partitioned data are separating sets of observations with all their features, for instance, the sales of different shops each offering the same items. By contrast, vertical partitioning means that each location senses a different subset of features. This is a common scenario in the Internet of Things [653]. Our steel rolling scenario is of this kind. It also appears, for instance, in traffic prediction problems (see also Section 4.1). Here, we might like to reduce communication costs by transmitting only aggregated label information between nodes. Again, the question is, if we can learn a model that is sufficiently accurate in assigning class labels to individual instances, based only on aggregated label information.

The problem of LLP not only deviates from that of supervised learning, where we learn from individually labeled training examples, but also from many other learning settings known in machine learning and data mining. It is different from *semi-supervised learning* [120], where we are given at least some examples that are labeled. It is not strictly *unsupervised learning*, since we are given at least *some* additional information about labels. It is different from *anomaly and outlier detection*, where we might know about observations that belong to a normal class. It comes close to *multiple instance learning* [731], where whole bags of observations are either labeled as positive or negative. However, LLP is not exactly the same problem, since we are not given binary information on each bag, but real-valued statistical information about the labels in each bag.

In Section 3.3.2, the task of LLP is defined more formally. Then, Section 3.3.3 gives an overview of related work. In Section 3.3.4, we discuss the difficulty of the problem from a Bayesian perspective. Afterwards, Section 3.3.5 defines loss functions for the scenario. Section 3.3.6 introduces a clustering approach and variants that minimize aforementioned loss functions. In Section 3.3.7, we compare the algorithm's prediction performance and runtime to other existing methods. Finally, Section 3.3.8 concludes and gives a short summary.

3.3.2 The Problem of Learning from Label Proportions

To the best of our knowledge, Musicant et al. were the first who formally formulated both the classification and regression tasks of the problem [466]. We extend their problem definition to multi-class problems and relate it to the unknown joint distribution $\mathbb{P}(\mathcal{X}, \mathcal{Y})$ from which all observations and labels are drawn.

Definition 1 (Learning from label proportions (LLP))**.** Let $\mathcal{X}$ be an instance space and $\mathcal{Y}$ be a space of categorical class labels $Y_1, \ldots, Y_l$. Let $\mathbb{P}(\mathcal{X}, \mathcal{Y})$ be an unknown joint distribution on the instances and class labels. In the setting of *learning from label*

Labeled examples (unknown)

$B_1 = \{(x_1, 1), (x_3, 1), (x_7, 0)\}$
$B_2 = \{(x_2, 0), (x_4, 0), (x_5, 1), (x_6, 1)\}$
$B_3 = \{(x_8, 0), (x_9, 0)\}$

Unlabeled examples (known)

$B_1 = \{x_1, x_3, x_7\}$ $n = 9$
$B_2 = \{x_2, x_4, x_5, x_6\}$ $h = 3$
$B_3 = \{x_8, x_9\}$ $l = 2$

Label proportions (known)

$Y = \{0, 1\}$

$$\begin{array}{cc} y=0 & y=1 \end{array}$$

$$\Pi = \begin{pmatrix} 0.33 & 0.67 \\ 0.50 & 0.50 \\ 1.00 & 0.00 \end{pmatrix} \begin{array}{l} |B_1| = 3 \\ |B_2| = 4 \\ |B_3| = 2 \end{array}$$

$$\eta \quad 0.56 \quad 0.44$$

Fig. 3.9: Example for given bags of observations, a label proportion matrix, and related notations

proportions, we are given a sample of N unlabeled observations $X = \langle x_1, \ldots, x_N \rangle$ with $X \subseteq \mathcal{X}$, drawn i.i.d. from $\mathbb{P}(\mathcal{X})$. Let $y_i \in Y$ be the individual class label for observation x_i, where $Y \subseteq \mathcal{Y}$. The individual labels are unknown. Instead, we are given a partitioning of X into h disjunct bags $B_1, \ldots, B_h$ and for each bag B_u and label Y_v, we are given the proportion $\pi_{uv} \in [0, 1]$ of observations with that label in bag B_u. Only based on this information, we seek a function (model) $\hat{f} : \mathcal{X} \to \mathcal{Y}$ that predicts the label $y \in \mathcal{Y}$ for an observation $x \in \mathcal{X}$ drawn i.i.d. from $\mathbb{P}$, such that the expected risk

$$R_{\text{exp}} = \int \ell(\mathcal{Y}, \hat{f}(\mathcal{X})) d\mathbb{P}(\mathcal{X}, \mathcal{Y})$$

is minimized. Here, ℓ is a convex *loss function* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$ which measures the cost of assigning the wrong label to individual observations.

The given label proportions π_{uv} can more conveniently be written as a $h \times l$ matrix $\Pi = (\pi_{uv})$, where the values in a row $\Pi_{u,\cdot} = (\pi_{u1}, \ldots, \pi_{ul})$ sum up to one. The frequency count μ_{uv} of observations with label $Y_v \in \mathcal{Y}$ in bag B_u can easily be reconstructed by multiplying the label proportion π_{uv} with bag size $|B_u|$.

The proportion $\eta(\Pi, Y_v)$ of label Y_v over the whole sample can then be calculated from Π as the sum of the frequency counts for bag u, divided by the total number of observations N:

$$\eta(\Pi, Y_v) = \frac{1}{N} \sum_{u=1}^{h} \mu_{uv} \,. \tag{3.2}$$

Figure 3.9 gives an example of the notations previously introduced, the division of observations into disjunct bags, and the label proportion matrix as derived from the original (now unknown) labels.

3.3.3 Related Work

When starting the work on LLP in 2010, only a few publications on the topic were available. In the following, we'll discuss related work insofar as it has been relevant for comparison with the clustering approach we published at that time [651] [652] and which is going to be presented in this contribution. Since its publication, our approach has been cited more than 70 times according to Google Scholar and even more works on the problem setting have appeared. For instance, different clustering methods have been applied [154]. Fish and Reyzin cast light on the theoretical properties of the problem in the context of probably approximately correct (PAC) learning [209] (for an easy introduction into PAC learning see Mitchell [447]). Saket et al. present an approach that does not rely on the underlying distributions of the bags, and give some guarantees for any learner [571]. However, they do not cover multiclass learning as we do here. Kobayashi and colleagues show estimation bounds for multiclass LLP [333]. A probabilistic method for LLP is developed to estimate the individual votes during presidential US elections [660]. Also new applications are dealt with such as bank customer classification [513], marine litter beaching prediction [273], and bankruptcy prediction [133].

Related Semi-Supervised Methods There are some approaches that seem similar to the scenario of LLP, but are actually semi-supervised learning tasks. For instance, Dara et al. first cluster the given data with SOMs and then label the resulting clusters [159]. However, labeled observations are given, which are usually not available when learning from label proportions. Demiriz et al. adapt the k-means optimization problem to respect labeled data [167]. Again, this is a semi-supervised setting, with labeled observations.

Basic Methods To the best of our knowledge, Kueck and Freitas were the first who introduced the problem of LLP by proposing a probabilistic model based on group statistics that is trained by an efficient Markov-Chain-Monte-Carlo (MCMC) sampling algorithm [357]. Musicant et al. were the first who defined the problem of learning from aggregate values for regression and classification tasks in a more formal way [466]. They modify well-known methods such as k-NN [13], backpropagation neural networks [447] and the linear SVM [700] to respect the given label proportions. Their experimental results focus on regression tasks, while we are mainly interested in classification.

Mean Map and Laplacian Mean Map The Mean Map method we use for comparisons in Section 3.3.7 has been proposed by Quadrianto et al. [514]. It estimates the conditional class probability $\mathbb{P}(\mathcal{Y}|\mathcal{X}, \vec{\theta})$ by conditional exponential models, using a joint feature map ϕ that maps observations and labels into a new feature space. The parameters $\vec{\theta}$ can be estimated by solving a convex maximization problem for the conditional

log-likelihood. The conditional log-likelihood in turn can be expressed in terms of the so-called *mean operator*, which can be expanded into bag-wise and label-wise components. The unknowns in this formulation can then be found by solving a system of linear equations, without knowing the individual labels. This is possible by making a *homogeneity assumption*, which states the conditional independence of feature vectors from bags, given the label. Once the mean operator is estimated, the parameter vector $\vec{\theta}$ can be derived by standard methods for maximum likelihood estimation. It is shown that Mean Map outperforms kernel density estimation, discriminative sorting, and MCMC [357].

Patrini et al. relax Mean Map's restrictive homogeneity assumption such that whenever bags are similar to each other, it is assumed that also their feature vectors are similarly distributed, given the label [494]. The relaxed assumption is encoded into a regularized least-squares minimization problem, which can be rewritten in matrix form by the Laplacian of a symmetric matrix whose entries consist of the similarities between bags. The solution to the stated optimization problem can then be obtained in closed form. On ten datasets from the UCI standard repository [32], LMM and AMM outperform Mean Map, Invcal, and the $\propto$SVM in terms of prediction performance and runtime. However, since LLM is not kernelized and can only find linear decision boundaries, the results and LLM cannot be directly compared with the non-linear clustering algorithm introduced in Section 3.3.6.

Inverse Calibration Rüping proposes the Inverse Calibration (Invcal) method [552]. The regression SVM (SVR) is converted into a probabilistic classifier by applying a scaling function σ to the outputs. According to the author, it is sufficient that the predictions of the classifier approximate the given label proportions well on average, for each bag. These constraints are integrated as auxiliary conditions into the standard SVR optimization problem. As a large margin method, the formulation allows for the reduction of model complexity, while the class probability estimates for each bag are kept close to the given label proportions for each bag, up to some maximum tolerable error. The primal problem can be transformed into its dual, and then solved with a standard solver for quadratic optimization. It is shown empirically over twelve standard datasets from the UCI repository that Invcal significantly outperforms Mean Map in terms of prediction accuracy.

$\propto$SVM Invcal treats the mean of each bag as some kind of super-instance, and gives each bag a regression label that corresponds to the label proportions. Instead, the $\propto$SVM proposed by Yu et al. explicitly models the labels of individual observations [751]. The label proportions, as calculated from labels assigned to individual observations, should match the given label proportions as closely as possible. This criterion is encoded as an additional term into the primal problem of the standard SVM. The task is to find a vector of labels such that the loss over label proportions and the standard loss

over individual observations are minimized. This ensures that observations lying on the same side of the hyperplane will be assigned the same label, depending on the particular values of the trade-off parameters C and C_π. Although the formulation seems intuitive, the optimization problem is a NP-hard non-convex integer programming problem. The authors propose two different efficient algorithms for solving it, one based on an alternating optimization strategy, and another based on convex relaxation. In experiments, the $\propto$SVM outperformed Mean Map and Invcal in terms of accuracy on several datasets from the UCI standard repository. However, the authors do not report which significance test they used. It should be noted that in the work by Patrini et al. results are not always in favor of the $\propto$SVM in comparison to Invcal, even on the same datasets [494]. Moreover, Mean Map outperformed the $\propto$SVM in many cases, while Invcal outperformed Mean Map in the work by Rüping [552].

AOC Kernel K-Means AOC Kernel k-means (AOC for Aggregate Output Classification) introduced by Chen et al., called AOC-KK in the following, clusters the observations such that clusters correspond to classes, and the assignment of observations to clusters (classes) matches the given label proportions [131]. The authors present variants of k-Means and kernel k-means [174], which is a kernelized version of the original k-means algorithm. Here, the cluster centers can no longer be written in explicit form, but have to be expressed in terms of a kernel function induced by some feature map ϕ.

In the objective function formulated, the first term is the same as in the original optimization problem of kernel k-means, while the second term measures the deviance between the given label proportions and those that would result from the current assignment of observations to clusters (classes). In that way, the authors try to find a good clustering, i.e. an assignment of observations to clusters (classes), such that the within-cluster scatter is minimized, but at the same time that also the given label proportions are matched as well as possible. The trade-off between the two criteria can be controlled by parameter λ. As standard tools for convex optimization cannot be used, the authors propose an alternating updating algorithm based on expectation maximization (EM) [168]. On two datasets from the UCI standard repository, AOC-KK outperforms k-NN and neural networks in terms of accuracy [466].

Although AOC-KK shares similarities with the clustering approach LLPC developed in Section 3.3.6, there are some fundamental differences. The first is that AOC-KK restricts the number of clusters to the number of classes, while LLPC allows for classes being represented by more than one cluster. This allows for a better control of bias vs. variance, by changing k. Another difference is that AOC-KK combines the loss over label proportions with the original kernel k-means objective in the same objective function, while LLPC first clusters observations as usual, and then tries to find a good assignment of labels to the resulting clusters. LLPC thus has the advantage that it can be used with arbitrary partitional clustering algorithms, while AOC-KK works only

with k-means and kernel K-means. LLPC is compared to AOC-KK with a quadratic loss function in Sect. 3.3.7.

Theoretical Results In an unpublished work, Yu et al. cast LLP into the PAC learnability framework [750]. The authors prove that under certain conditions, the labels of individual observations can be predicted well when the label proportions per bag can be predicted well. The generalization error over bag proportions in turn can be bounded by the empirical proportion error if the number of bags is large in relation to the Vapnik Chervonenkis (VC) dimension of the underlying hypothesis class $\mathcal{H}$ [1]. The authors further show that the probability for classifying instances correctly increases with the purity of bags, i.e. if many instances per bag belong to the same class. In extreme cases, where all label proportions are equal (i.e. they are the least pure), it can happen instead that a hypothesis achieves zero bag proportion error, but nevertheless classifies all instances incorrectly. The true error can be even further bounded by making additional assumptions on the distribution of bags [750].

The aforementioned findings imply that special care must be taken when comparing the performance of label proportion learning methods. For instance, it must be ensured that algorithms are trained and validated on the exact same data splits. Moreover, since the individual bag distributions can play a big role for the performance of algorithms, performance should be assessed over different diverse datasets and results need to be tested for their significance, even more so than with traditional supervised methods. That a method outperforms another does not mean then that it shows better performance in an absolute sense, under all circumstances, but on average.

Other Works Hernández-González et al. apply a structural EM strategy to learn Bayesian network classifiers from label proportions [274]. They compare their method to Mean Map and report lower error rate of their method for four of seven domains. However, the significance of results is not reported. Fan et al. introduced a generative classifier called DNLP, which learns from label proportions by following a deep belief network approach [198]. The authors compare their method to Mean Map and Invcal on several standard datasets from the UCI repository. In terms of prediction performance, they report no significant differences. However, the runtime of DNLP is much lower than that of Mean Map and Invcal. Fan and Taylor combine convolutional neural networks (CNN) with probabilistic graphical models trained by an EM approach to learn from label proportions in the context of ice and open water classification from image data [199]. Their algorithm shows good performance in the context of the mentioned application, but isn't evaluated on other domains.

1 For an easy introduction we recommend Mitchell [447] explaining PAC learning and the VC dimension.

3.3.4 Difficulty of the Problem

For getting a better idea about the difficulty of the problem, we discuss the problem of LLP from a more Bayesian perspective and relate it to different kinds of better-known learning tasks such as supervised, semi-supervised, and unsupervised learning.

For the supervised learning scenario, we can construct an optimal classifier, called the *optimal Bayes classifier*, if the distribution $\mathbb{P}(\mathcal{X}, \mathcal{Y})$ is known. From a Bayesian perspective, a prediction model can be obtained from estimating the conditional class density $\mathbb{P}(\mathcal{Y} \mid \mathcal{X})$. Applying Bayes theorem, one recognizes that $\mathbb{P}(\mathcal{Y} \mid \mathcal{X})$ may also be estimated from other unknown densities—the class-conditional density $\mathbb{P}(\mathcal{X} \mid \mathcal{Y})$ and the class prior density $\mathbb{P}(\mathcal{Y})$:

$$\mathbb{P}(\mathcal{Y} \mid \mathcal{X}) = \frac{\mathbb{P}(\mathcal{X} \mid \mathcal{Y}) \cdot \mathbb{P}(\mathcal{Y})}{\mathbb{P}(\mathcal{X})} \tag{3.3}$$

Here, $\mathbb{P}(\mathcal{X})$ doesn't necessarily need to be known or estimated, since it can be calculated from $\mathbb{P}(\mathcal{X} \mid \mathcal{Y})$ and $\mathbb{P}(\mathcal{Y})$. $\mathbb{P}(\mathcal{Y})$ may be estimated directly from the data, if the number of data points is high enough. Moreover, if the joint distribution $\mathbb{P}(\mathcal{X}, \mathcal{Y})$ is known, as is assumed by the optimal Bayes classifier, all other quantities can be derived from it. For a given observation $x \in \mathcal{X}$ to classify, the optimal Bayes classifier would predict the most probable class, which is also known as the MAP criterion. Here, optimal means that the Bayes classifier is the best classifier over all possible classifiers for the given data.

Best Case With respect to LLP, the class prior $\mathbb{P}(Y_v)$ for label Y_v can be estimated as $\eta(\Pi, Y_v)$, the proportion of Y_v. This is done, for instance, by the Mean Map method. Finding a good estimate for $\mathbb{P}(\mathcal{X} \mid \mathcal{Y})$, however, is at least as difficult as in the supervised scenario and equates to it if each bag B_u only contains observations from a single class and at least l bags contain observations from different classes. This scenario may be called the *best case*, since in LLP, usually less information about individual labels is given. Our intuition matches the findings of Yu et al. where it has been proven that the probability of classifying instances correctly increases with the purity of the bags [750]. When each bag only contains examples from the same class, each bag is as pure as possible.

Worst Case In the *worst case*, all label proportions π_{uv} in matrix Π are equal, i.e. least pure, and labels can only be guessed correctly with probability $1/l$. If sample size is large enough, the worst case can only occur if also all class priors $\mathbb{P}(Y_v)$ are equal, and the label does not depend on the bag, i.e. $\mathbb{P}(\mathcal{Y}|u) = \mathbb{P}(\mathcal{Y})$. Otherwise, we can estimate $\mathbb{P}(\mathcal{Y})$ from the data and at least predict the class that has highest probability to occur (i.e. the majority class). In this case, the probability for predicting the correct label would be higher than $1/l$.

The worst case can only occur with large amounts of data, where the label proportions, given as relative frequencies, will approach the true probabilities of classes in each bag, or in the context of privacy-preserving data mining, where we have full control over the formation of bags and want to make the problem as difficult as possible. In general, a small number of large sized bags can make the problem more difficult, as Yu et al. have shown [750]. For smaller random samples, we may usually expect slight deviations of the proportions in matrix Π, which may help with making a correct decision about class labels. For instance, when randomly uniformly sampling the 50 observations per class from the well-known Iris dataset into bags, in many cases the clustering approach developed in Section 3.3.6 classifies at least 96 % of the observations correctly on average.

Average Case In cases where observations have been sampled more or less randomly into bags, a first intuition might be that bags that are more "pure", i.e. that contain more instances of the same class, provide more information. Yu et al. also show that the probability of classifying individual instances correctly increases with the purity of bags [750]. However, only an upper bound is shown for the *probability* of misclassifying a fraction of individual observations incorrectly. In practice, cases may occur where we perform well, despite label proportion matrix Π having high entropy, or badly, despite Π having low entropy.

For instance, bags with low information content in terms of labels may nevertheless provide information about the underlying distribution of observations, $\mathbb{P}(\mathcal{X})$. Getting more information about $\mathbb{P}(\mathcal{X})$ by taking unlabeled observations into account as well can increase prediction performance when only a few labeled examples are given [120]. Conversely, even if a bag has high information content in terms of labels, learning might not profit from it if the sample doesn't represent the underlying data distribution.

For practical cases, it is therefore hard to find a measure of problem difficulty. While it is easy to measure the entropy of Π, it is difficult to measure how well bags reflect the overall data distribution given a concrete sample, without knowing the underlying data distribution—which is the crux of learning.

3.3.5 Loss and Risk

There are different possible ways to define loss functions over label proportions. First we define measures of the quadratic deviation between the label proportions as being derived from a previously trained prediction model $\hat{f}$ and the given label proportions. Applying the trained model to a set of observations $x_i \in X$, the resulting label proportions can be calculated by counting the number of observations x_i with $\hat{f}(x_i) = Y_v$, in each bag for each label $Y_v \in \mathcal{Y}$ and dividing such counts by the size of their respective

bag. This leads to a new matrix $\Gamma_{\hat{f}}$, containing the model-based label proportions:

$$\Gamma_{\hat{f}} = (\gamma_{uv}^{\hat{f}}), \quad \gamma_{uv}^{\hat{f}} = \frac{1}{|B_u|} \sum_{x \in B_u} I(\hat{f}(x), Y_v), \quad I = \begin{cases} 1 : \hat{f}(x) = Y_v \\ 0 : \hat{f}(x) \neq Y_v \end{cases} . \tag{3.4}$$

Similar to when defining a loss function for individual observations, it is now possible to define a loss function over individual matrix entries by say, taking as loss the squared error $(\pi_{uv} - \gamma_{uv}^{\hat{f}})^2$. The total deviance between Π and $\Gamma_{\hat{f}}$ can then be defined as the average squared error over all matrix entries:

$$\ell_{MSE}(\Pi, \Gamma_{\hat{f}}) = \frac{1}{hl} \sum_{u=1}^{h} \sum_{v=1}^{l} (\pi_{uv} - \gamma_{uv}^{\hat{f}})^2 \tag{3.5}$$

The average squared error ℓ_{MSE} doesn't take into account the relative group and class sizes; nor can it catch the situation where two hypotheses $\hat{f}_1$ and $\hat{f}_2$ appear indistinguishable from each other, because the total error sum over all matrix entries is the same. In practice, it can make sense to measure the error between Π and $\Gamma_{\hat{f}}$ by ℓ_{Π}, which we define as the geometric mean of two different error measures ℓ_{weight} and ℓ_{Prior} which deal with the previously mentioned disadvantages:

$$\ell_{\Pi}(\Gamma_{\hat{f}}) = \sqrt{\ell_{weight}(\Pi, \Gamma_{\hat{f}}) \cdot \ell_{Prior}(\Pi, \Gamma_{\hat{f}})} \quad \text{with} \tag{3.6}$$

$$\ell_{weight}(\Pi, \Gamma_{\hat{f}}) = \frac{1}{hl} \sum_{u=1}^{h} \sum_{v=1}^{l} \eta(\Pi, Y_v) \frac{|B_u|}{N} (\pi_{uv} - \gamma_{uv}^{\hat{f}})^2 \quad \text{and} \tag{3.7}$$

$$\ell_{Prior}(\Pi, \Gamma_{\hat{f}}) = \frac{1}{l} \sum_{v=1}^{l} \Big(\eta(\Pi, Y_v) - \eta(\Gamma_{\hat{f}}, Y_v) \Big)^2 \tag{3.8}$$

ℓ_{weight} weights the squared error of individual matrix entries by their relative group and class size. ℓ_{Prior} measures how well a chosen hypothesis matches the class priors, as estimated by $\eta(\Pi, Y_v)$. The choice to include the prior in the loss function has been made based on empirical evaluations and a close examination of the label proportion matrices which have lead to misclassifications. What we have observed in our experiments now has a theoretical justification. As shown by Yu et al., whenever a hypothesis matches the class priors and observations in bags are distributed i.i.d., the probability of misclassfiying a fraction of individual observations is bounded [750].

Moreover, if in addition to the label proportions, the true labels $y(x)$ of a subset $T \subseteq X$ of observations $x \in T$ are given, error criterion (Equation 3.6) can be easily extended to include the average loss ℓ_T over these labeled training examples:

$$\ell_{\Pi} = \sqrt[3]{\ell_{weight} \cdot \ell_{Prior} \cdot \ell_T} \quad \text{with} \quad \ell_T = \frac{1}{|T|} \sum_{x \in T} \ell(y(x), \hat{f}(x)) \tag{3.9}$$

Algorithms that optimize over ℓ_{Π} can thereby easily consider also labeled observations in addition to the given label proportions.

3.3.6 Learning from Label Proportions by Clustering

The goal in LLP is to find a function $\hat{f}$ that predicts the proportions of previously unseen bags as well as possible, which in turn bounds the risk of misclassifying individual observations, as shown by Yu et al. [750]. The authors pose this problem in terms of empirical risk minimization. However, if we allow for arbitrarily complex hypotheses, we can always match the given label proportions. In particular if we tried all different possible labelings of observations exhaustively, we would always find a set of labelings that minimizes one of the previously introduced loss functions. We would expect only a few of such labelings to also minimize the empirical loss over individual observations, i.e. we somehow need to control the capacity of our hypothesis class.

The particular LLP approach proposed in the following is based on the assumption that observations lying close together in regions of the input space also share the same class label. It first forms clusters of similar observations using an arbitrary partitional clustering algorithm and respective distance measure. Instead of trying all possible labelings of observations, the algorithm heuristically tries different labelings of clusters, such that a loss function over label proportions is minimized. The capacity of the hypothesis space can thus be controlled by varying the number of clusters k. A small number of clusters leads to high bias, but low variance. A larger number of clusters allows for ever smaller divisions of sample X, and therefore leads to low bias, but high variance.

The assumption that clusters represent classes is not necessarily correct. Hastie et al. demonstrate that especially the weighting of features can have an enormous influence on clustering results [259]. In fact, one advantage of supervised methods over unsupervised ones is that they can determine the relevance of features in relation to the target variable. We therefore allow for a certain flexibility in distance measures. Such measures should respect weights $w_j \in [0, 1]$ for each feature A_j, as given by a vector $\vec{w} = (w_1, \ldots, w_d)$. Usually, such weights are specified by a domain expert. In the clustering approach introduced in the following, however, the relevance weights can be approximated automatically by an evolutionary strategy, based on one of the loss functions defined in Section 3.3.5 (or other loss functions for LLP).

In the Section 3.3.6.1, the accompanying optimization problem is stated. Then, in Section 3.3.6.2, an approach for solving it is described. The algorithm can be used with different labeling strategies which are presented in Section 3.3.6.3. The approach's runtime is analysed in Section 3.3.6.4, while Section 3.3.6.5 explains how to classify new examples, based on a set of labeled clusters.

3.3.6.1 Optimization Problem

Let the vector $\vec{\lambda}_{\mathcal{C}} = (\lambda_1, \ldots, \lambda_k)$ with $\lambda_j \in \mathcal{Y}$ represent a labeling for a clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$. Let $\hat{f}_{\vec{\lambda}_{\mathcal{C}}} : \mathcal{X} \to \mathcal{Y}$ be a mapping that returns the label λ_i for a given observation $x \in C_i$. Given a clustering $\mathcal{C}$, we search for a labeling $\vec{\lambda}_{\mathcal{C}}$ of the clusters

that minimizes the error, according to some error measure $\ell_{\vec{\lambda}_\mathcal{C}}(\Pi, \Gamma_{\hat{f}_{\vec{\lambda}_\mathcal{C}}})$, between the model-based label proportions $\Gamma_{\hat{f}_{\vec{\lambda}_\mathcal{C}}}$ and the known label proportions Π:

$$\min_{\vec{\lambda}_\mathcal{C}} \ell_{\vec{\lambda}_\mathcal{C}}(\Pi, \Gamma_{\hat{f}_{\vec{\lambda}_\mathcal{C}}}) \tag{3.10}$$

The error measure could be, for instance, the average squared error ℓ_{MSE} or a combined error measure such as ℓ_Π.

Let $q_{\vec{w}}$ be a function which is able to assess the quality of a clustering based on a similarity measure that respects feature weights. This usually means that observations $x_i \in \mathcal{X}$ are represented as d-dimensional feature vectors $\vec{x}_i = (x_{i1}, \ldots, x_{id})$ with $x_{ij} \in \mathbb{R}$. We are trying to solve the optimization problem

$$\min_{\vec{w}} \ell_{\vec{\lambda}_\mathcal{C}}(\Pi, \Gamma_{\hat{f}_{\vec{\lambda}^\star_\mathcal{C}}}), \quad \vec{\lambda}^\star_\mathcal{C} = \operatorname{argmin}_{\vec{\lambda}_{\mathcal{C}^\star}} \ell_{\vec{\lambda}_\mathcal{C}}(\Pi, \Gamma_{\hat{f}_{\vec{\lambda}_{\mathcal{C}^\star}}}), \quad \mathcal{C}^\star = \operatorname{argmax}_\mathcal{C} q_{\vec{w}}(\mathcal{C}), \tag{3.11}$$

i.e. we are searching for a clustering $\mathcal{C}^\star$ which maximizes $q_{\vec{w}}$ and whose labeling $\vec{\lambda}^\star_\mathcal{C}$ minimizes $\ell_{\vec{\lambda}_\mathcal{C}}$, for all possible weight vectors $\vec{w}$. As formulated, with arbitrary functions $q_{\vec{w}}$ and $\ell_{\vec{\lambda}_\mathcal{C}}$, the problem is non-convex. Since we want to allow for flexibility in the choice of such functions, in the following we approximate solutions by an evolutionary strategy.

3.3.6.2 The LLPC Algorithm

The LLPC (Learning from Label Proportions by Clustering) algorithm solves problem (3.11) by an evolutionary strategy. For each weight vector $\vec{w}$, the sub-optimization problem of maximizing $q_{\vec{w}}$ is solved by an inner clustering algorithm, where the particular $q_{\vec{w}}$ depends on the algorithm. The only prerequisite for the clusterer is that it returns disjunct clusters and respects different feature weights. The sub-optimization problem (3.10) is independent from the clusterer and currently can be solved by different labeling strategies, of which two are introduced in Section 3.3.6.3.

In more detail, LLPC takes a clustering algorithm *clusterer*, a labeling algorithm *labeler* and an error measure $\ell_{\vec{\lambda}_\mathcal{C}}$ as parameters, in addition to Π, X, $\mathcal{B} = \{B_1, \ldots, B_h\}$ and $Y = \{Y_1, \ldots, Y_l\}$, which are related to the task of LLP, and a set of parameters *evo* related to the evolutionary learning strategy. LLPC then approximates the optimal weight vector and returns $\vec{w}^\star$, as well as the related clustering $\mathcal{C}^\star$ and labels $\vec{\lambda}^\star_\mathcal{C}$ for the clusters. The returned weights $\vec{w}^\star$ can be interpreted as the importance of individual features and thus give valuable additional information for the interpretation of cluster models.

We use the evolutionary strategy described in [444]. The evolutionary strategy starts with a random population P of normalized weight vectors $\vec{w}$, i.e. $w_j \in [0, 1]$. For each individual in P, the clustering algorithm *clusterer* is called. The clusters are labeled according to the given labeling algorithm *labeler* and the fitness is evaluated by criterion $\ell_{\vec{\lambda}_\mathcal{C}}$. If the fitness is higher than the best fitness seen so far, the newly found

Algorithm 1: The LLPC algorithm

Function LLPC(Π,X,$\mathcal{B}$,$\mathcal{Y}$,*clusterer*,k, *labeler*,$\ell_{\vec{\lambda}_\mathcal{C}}$,*evo*)
  best_fit := $-\infty$; *generation* := 0 ;
  Randomly initialize a population P of *psize* normalized weight vectors ;
  while *generation* < *maxgen* **do**
    for $\vec{w} \in P$ **do**
      $\mathcal{C}$:= *clusterer*(X, k, $\vec{w}$) ;
      ($\vec{\lambda}_\mathcal{C}$, *err*) := *labeler*($\mathcal{C}$, $\mathcal{B}$, Π, $\mathcal{Y}$, $\ell_{\vec{\lambda}_\mathcal{C}}$);
      if *best_fit* < $-$*err* **then**
        best_fit := $-$*err*; $\mathcal{C}^\star$:= $\mathcal{C}$; $\vec{\lambda}^\star_\mathcal{C}$:= $\vec{\lambda}_\mathcal{C}$; $\vec{w}^\star$:= $\vec{w}$;
      end
    end
    generation := *generation* + 1 ;
    if *generation* < *maxgen* **then**
      Pcopy := P ;
      Gaussian mutation of weights in *Pcopy* with variance *mutvar* ;
      Pchildren := Uniform crossover on $P \cup$ *Pcopy* with probability *crossprob* ;
      P := Tournament selection with size *tournsize* on $P \cup$ *Pcopy* $\cup$ *Pchildren* ;
    end
  end
  return $\mathcal{C}^\star$, $\vec{\lambda}^\star_\mathcal{C}$, $\vec{w}^\star$;
end function

clustering, labeling, and weight vector are memorized as the new best ones. In each generation, the weight values in a copy of P are mutated by a Gaussian distribution and, with a certain probability, exchanged with P by a crossover operator. The individuals then take part in a tournament and only the best ones are kept in the next generation. This process is repeated until the maximum number of generations as specified by the user is reached.

Using an evolutionary strategy as a wrapper has the advantage that it is not necessary to integrate the error measure $\ell_{\vec{\lambda}_\mathcal{C}}$ into the optimization problem of the inner clustering algorithm, as was done in AOC-KK. The clustering algorithm can thus be treated as a black box and easily exchanged, without any further adaptation. It should also be noted again that in contrast to AOC-KK, LLPC allows for classes being represented by more than just one cluster ($k > l$). Thereby LLPC allows for ever smaller divisions of sample X, i.e. parameter k may be seen as a control parameter that trades off bias against variance, as previously discussed.

The free choice of clustering algorithm allows for respecting different kinds of data distributions. For example, LLPC was run successfully with k-means [426], kernel k-means [174], EM clustering [168, 731], DBSCAN [191], PROCLUS [11] and Support Vector Clustering (SVC) [55], without modification. Moreover, LLPC can be used with different

error measures, such as criterion (Equation 3.9) that can respect individually labeled examples.

LLPC may therefore be looked at as a meta-algorithm for learning from label proportions, which allows for the use of different clustering algorithms, labeling strategies and loss functions. In a further step, one might also exchange the evolutionary algorithm. For instance, it might be adapted to not only minimize $\ell_{\vec{\lambda}_{\mathcal{C}}}$ over weight vector $\vec{w}$, but also over hyperparameters such as k in the case of k-means clustering, or C and the RBF kernel γ in the case of SVC.

Algorithm 2: Labeling of clusters by local search with multistarts

Function LocalSearchMultiStart($\mathcal{C}, \mu, \Pi, k, \mathcal{Y}, \ell_{\vec{\lambda}_{\mathcal{C}}}, starts$)
  $best = -\infty$;
  for $iteration \leftarrow 1, starts$ **do**
    $\vec{\lambda}_{\mathcal{C}}, \vec{\lambda}_{\mathcal{C}}^{\text{bestIter}} \leftarrow (\lambda_1, \ldots, \lambda_k)$ with $\lambda_j \in \mathcal{Y}$ chosen uniformly at random
    $start, bestIter \leftarrow -\ell_{\vec{\lambda}_{\mathcal{C}}}(\Pi, \Gamma_{\hat{f}_{\vec{\lambda}_{\mathcal{C}}}})$; *// calculate initial fitness*
    $improving \leftarrow$ true ;
    while *improving* **do**
      for $kpos \leftarrow 1, k$ **do**
        // at each position . . .
        for $lpos \leftarrow 1, |\mathcal{Y}|$ **do**
          // . . . try all labels . . .
          $\lambda_{kpos} \leftarrow Y_{lpos} \in \mathcal{Y}$;
          $fitness \leftarrow -\ell_{\vec{\lambda}_{\mathcal{C}}}(\Pi, \Gamma_{\hat{f}_{\vec{\lambda}_{\mathcal{C}}}})$ *// calculate fitness*
          if $fitness > bestIter$ **then**
            $\vec{\lambda}_{\mathcal{C}}^{\text{bestIter}} \leftarrow \vec{\lambda}_{\mathcal{C}}$; $start, bestIter \leftarrow fitness$;
            break *// leave both for loops*
          else
            $\lambda_{kpos} \leftarrow \lambda_{kpos}^{\text{bestIter}}$ *// reset to best label found at kpos so far*
          end
        end
      end
      if $bestIter = start$ **then**
        // Nothing better found
        $improving \leftarrow$ false ;
      end
    end
    if $bestIter > best$ **then**
      $best \leftarrow bestIter$; $\vec{\lambda}_{\mathcal{C}}^{\text{best}} \leftarrow \vec{\lambda}_{\mathcal{C}}^{\text{bestIter}}$ *// remember best solution*
    end
  end
  return $\vec{\lambda}_{\mathcal{C}}^{\text{best}}, -best$
End Function

3.3.6.3 Labeling Strategies

The following two labeling algorithms solve the sub-optimization problem (3.10) and can be used as the *labeler* in LLPC.

Exhaustive Labeling As long as k can be restricted to a small number and $l = 2$ for a binary classification problem, trying l^k possible labelings for a clustering $\mathcal{C}$ is no problem. In experiments (see Section 3.3.7), good solutions often were found for $6 \leq k \leq 12$. For each labeling, we need to calculate $\text{err}_{\vec{\lambda}_\mathcal{C}}$. For error measures like ℓ_{MSE} or ℓ_{Π}, this takes linear time in the number of observations N. In case of the aforementioned error measures, the calculations only involve basic operations such as count, addition, multiplication, and division.

Local Search with Multistarts For cases where the number of clusters $k > 12$ or the number of labels $l > 2$, a local search that is started multiple times with different random combinations of labels is proposed. The local search greedily improves on the current labeling of clusters by trying all possible labels at each component of the labeling vector $\vec{\lambda}_\mathcal{C}$. Fitness measures how well the model-based label proportion matrix $\Gamma_{\hat{f}}$, as calculated from the current labeling, matches the given label proportions Π. If the fitness improves, the search starts again from the first component of the labeling vector $\vec{\lambda}_\mathcal{C}$. Otherwise, it resets the label at the current position $kpos$ to the label of the best (local) solution found so far. The best labeling found over all starts of the different greedy searches is returned.

In each iteration, the greedy search runs until no further improvement is possible (which is a stopping criterion). Moreover, at each step of the algorithm, the fitness either improves or stays the same. Therefore, each search finds a local minimum. Since the number of searches is finite, the returned labeling vector is also locally minimal. Although, in contrast with the exhaustive labeling strategy, it cannot be guaranteed that a globally optimal solution will be found, it has been demonstrated that the heuristic labeling strategy performs well in real-world applications such as reducing communication costs in distributed machine learning applications [655].

3.3.6.4 Runtime Analysis

The user-specified parameters *maxgen*, *psize* and *tournsize* in LLPC are constants. They do not depend on the number of observations N and limit the number of iterations of the evolutionary strategy to be constant. As discussed in Section 3.3.6.3, the asymptotic runtime of the labeling strategies is linear in N, as k and l are constants and the evaluation of $\text{err}_{\vec{\lambda}_\mathcal{C}}$ usually takes linear time. The asymptotic runtime of LLPC will otherwise depend on the used cluster algorithm. For example, if we allow for approximate solutions and limit the number of iteration steps, k-means has a linear runtime. Hence, overall LLPC has linear runtime, which makes it especially well suited for use in resource-constrained settings such as applications in the Internet of Things settings.

However, when used with an algorithm like kernel k-means, the runtime of LLPC can also become quadratic, for instance.

3.3.6.5 Generating a Prediction Model

The LLPC algorithm returns labeled clusters of sample X. It is then possible to assign labels to individual observations $x_i \in \mathcal{X}$ with $\hat{f}_{\vec{\lambda}_e}$. The question is how to predict the labels of new observations, i.e. how to transform a clustering into a prediction model.

In the case of clustering algorithms which return a model-based description of clusters such as k-means which returns cluster means, one can simply use the model to assign new observations to a cluster, and then predict the cluster's corresponding class label. For instance, in the case of k-means, one can assign new observations to their closest cluster mean and predict the corresponding class label, by applying function $\hat{f}_{\vec{\lambda}_e}$. Whenever a clustering algorithm is purely descriptive, i.e. in cases where it only returns a clustering of X, but no model to assign new observations to clusters, one may use a nearest neighbors approach such as k-NN for classification.

In general, one option for getting a prediction model after running LLPC is to train a standard classifier such as Naïve Bayes [301] or a SVM [700], based on the current labeled observations. Taking this approach, LLPC may be regarded as a preprocessing step before modeling, in which the missing labels of observations in sample X are restored, based on the given label proportions.

3.3.7 Evaluation

In this section, we evaluate the general method. Motivated by the steel scenario, the method needs to be carefully evaluated in order to be trustfully applicable in diverse industrial applications. Since the method is general, the LLPC algorithm is compared with three state-of-the-art methods for LLP: the Mean Map [514] method, Inverse Calibration (Invcal) [552], and AOC Kernel k-Means (AOC-KK) [131]. The comparisons are performed using standard benchmark datasets or generated test data, as is usually done.

LLPC is written in Java and has been implemented in the form of several operators in RapidMiner (https://rapidminer.com). All results are based on using fast k-means [187] as an inner clustering operator, which is a variant of k-means utilizing the triangle inequality for faster distance calculations. Observations $x_i \in \mathcal{X}$ are represented as d-dimensional feature vectors $\vec{x}_i = (x_{i1}, \ldots, x_{id})$ with $x_{ij} \in \mathbb{R}$. As a distance measure we have used the weighted Euclidean distance with weight vector $\vec{w} = (w_1, \ldots, w_d)$

$$d_{\vec{w}}(\vec{x}_i, \vec{x}_i^{'}) = \sum_{j=1}^{d} (w_j x_{ij} - w_j x_{ij}^{'})^2 \tag{3.12}$$

Tab. 3.3: UCI datasets used for the experiments. ©[2011] Springer. Reprinted, with permission, from [651].

Dataset	*N*	*d*	Dataset	*N*	*d*
CREDITA	690	42	SONAR	208	60
VOTE	435	16	DIABETES	768	8
COLIC	368	60	BREAST CANCER	286	38
IONOSPHERE	351	34	HEARTC	303	22

which weights each feature x_{ij} of example $\vec{x}_i$ differently. In all experiments, we used the exhaustive labeling strategy (see Section 3.3.6.3) with loss function ℓ_Π (see Section 3.3.5). AOC-KK has been implemented using a combination of Java, RapidMiner, and Matlab. For Mean Map and Invcal, R scripts were used, which were provided by the author of Invcal [552].

3.3.7.1 Prediction Performance Experiments

The accuracy of LLPC, AOC-KK, Invcal, and Mean Map has been assessed on the eight UCI [32] datasets shown in Table 3.3. Each possible value of a nominal feature has been mapped to a binary numerical feature with values 0 or 1. Numerical features were normalized to the [0, 1] interval. Table 3.3 shows the number of features d after this preprocessing step.

In each single experiment, the accuracy has been assessed by a 10-fold cross-validation. For LLP, we have partitioned the training set of a particular fold into bags of size σ, by uniform sampling of observations. While such uniform sampling might not reflect the way in which bags are formed in a real-world setting, it allows for a more homogeneous interpretation of results across different datasets than domain-specific sampling based on feature values. We tried several bag sizes σ: 2, 4, 8, 16, 32, 64, and 128 (with the last bag smaller than σ, if necessary). The label proportions were calculated and the individual labels removed. In each fold, the accuracy of the learned prediction model has then been calculated on a labeled test set.

The kernel methods Mean Map, Invcal, and AOC-KK have been tested with the linear kernel, polynomial kernels of degree 2 and 3, and radial basis kernels ($\gamma = 0.01$, 0.1 and 1.0). LLPC has been tested for cluster sizes $k \in [2, 12]$. As parameters for the evolutionary strategy, we used $maxgen = 10$, $psize = 25$, $mutvar = 1.0$, $crossprob = 0.3$ and $tournsize = 0.25$. Running LLPC with k-means provides a prediction model consisting of cluster means with associated class labels. The same is true for AOC-KK. However, the cluster methods also assign labels to each observation in sample X, allowing for a subsequent training of other classifiers, as described in Section 3.3.6.5. Based on such labeled examples, we have trained models for Naïve Bayes [301], k-NN [13], decision trees [516], random forests [102], and the SVM [700] with linear and

radial basis kernel. The model parameters of each method have been optimized by a grid or evolutionary search.

The combination of all datasets, bag sizes, classifiers, their variants and parameters results in a total of 13 216 experiments: 672 for Mean Map and Invcal, 2688 for AOC-KK, and 9856 for LLP. For bag sizes 16, 32, 64 and 128 on the datasets COLIC and SONAR, and for bag size 128 on CREDITA, we conducted additional experiments with LLP for $maxgen = 5$ and $psize = 100$. In some cases, we achieved better prediction accuracy. All experiments took about three weeks. They were run in parallel on up to six machines with an AMD Dual-Core or Quad-Core Opteron 2220 processor and a maximum of 4 GB main memory.

3.3.7.2 Prediction Performance Results

Figure 3.10 contains plots of the highest achieved accuracies for all datasets and bag sizes, based on the best performing models of LLPC, AOC-KK, Invcal, and Mean Map, over all conducted experiments. LLPC shows a higher accuracy than Invcal for many bag sizes on the datasets CREDITA, VOTE, COLIC, SONAR, and BREAST CANCER. On CREDITA, VOTE, IONOSPHERE, SONAR, and DIABETES, the variance of accuracy between bag sizes is smaller for LLPC compared with the other methods. Mean Map performs worse than LLPC and Invcal in many cases. The performance of AOC-KK varies, depending on the dataset. It shows good performance on BREAST CANCER and HEARTC, but not on the others. Except for the BREAST CANCER and VOTE datasets and a few other accuracy values, the overall accuracy of *all* methods decreases with a larger bag size. The results thus confirm the theory: with larger sizes of bags, without increasing the size of sample X, learning becomes more difficult.

The statistical significance of results can be assessed with the adjusted version of the Friedman test, as proposed by [169]. The test is a non-parametric equivalent of ANOVA and ranks the classifiers for each dataset separately. Under the null-hypothesis, the average ranks of the classifiers should be equal. For comparing LLPC to all others, we proceeded with the two-tailed Bonferroni-Dunn test as a post-hoc test.

Table 3.4 can be understood as a summary of the detailed plots shown in Figure 3.10, giving a better understanding and overview of LLPC's overall performance. The table shows the average ranks of the compared classifiers and their difference to LLPC's rank. Each rank was calculated based on the best performing models (including the standard classifiers), over all conducted experiments. The table also shows the critical difference (CD) values for the Bonferroni-Dunn test. The CD for $\sigma = 128$ is different, because Mean Map was not included in the comparison, due to missing values. LLPC has the highest rank in six cases, for $\sigma > 2$. At the 10 %-level, LLPC is significantly better than AOC-KK for $\sigma = 8$, better than Invcal for $\sigma = 128$ and better than Mean Map for $\sigma = 4, 8, 32$, and 64. In all other cases, LLPC performs equivalently.

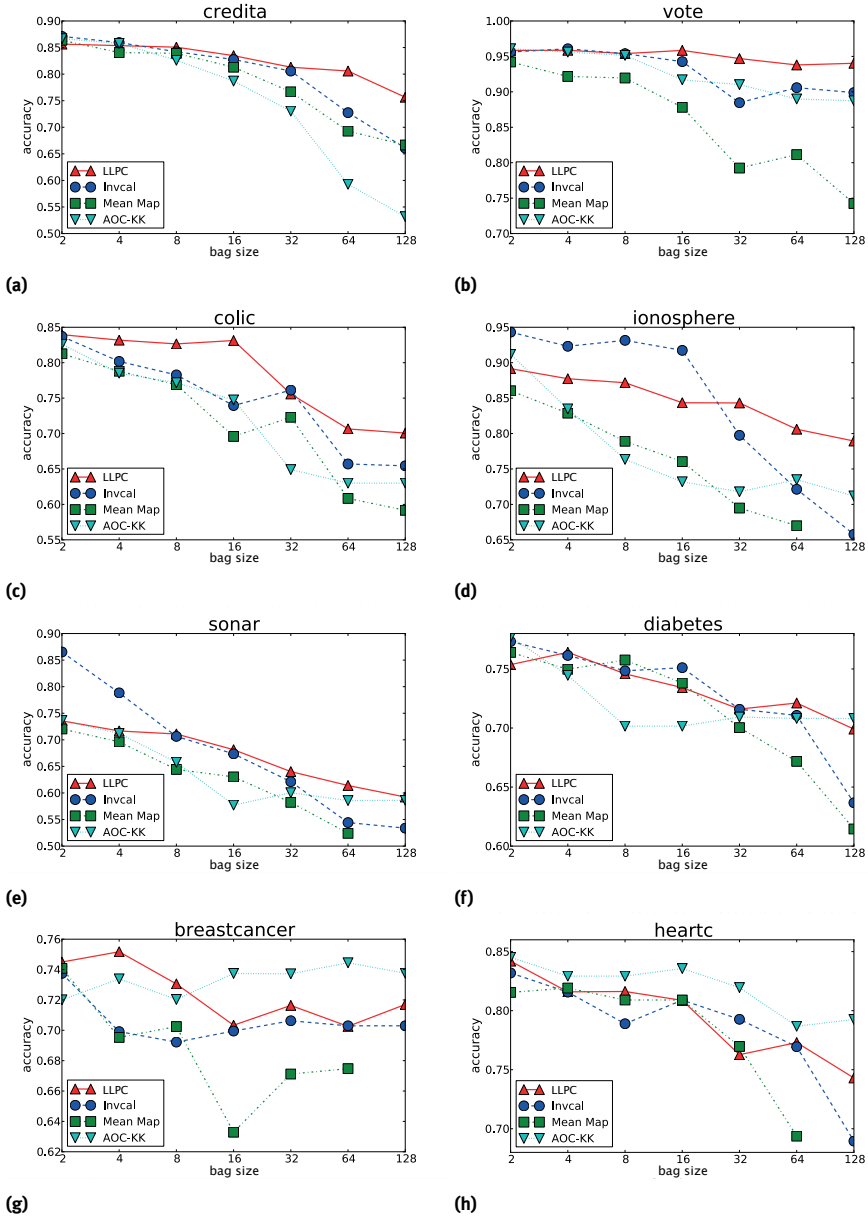
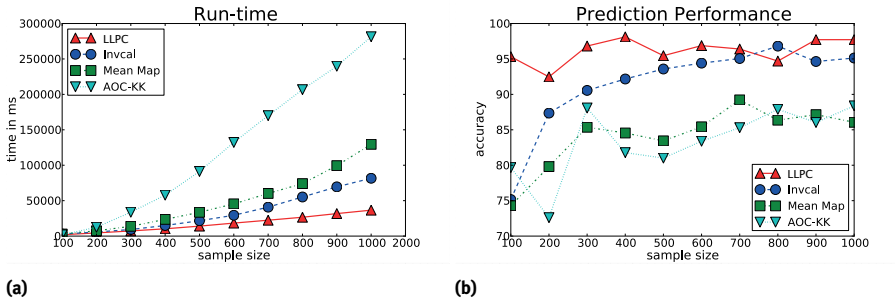


Fig. 3.10: Highest accuracies for all datasets and bag sizes, over all 13 216 runs of LLPC, AOC-KK, Invcal, and Mean Map (plus the additional runs of LLPC with $maxgen = 5$ and $psize = 100$). Some values for Mean Map and bag size 128 are missing in the plots, due to an error in the R script. ©[2011] Springer. Reprinted, with permission, from [651].

Tab. 3.4: Average ranks of classifiers by bag size, and their difference to LLPC's rank, based on the best models for each dataset and bag size. Positive difference values indicate a better performance of LLPC. Highest ranks and significant differences (higher than CD) at the 10 %-level are marked in bold. ©[2011] Springer. Reprinted, with permission, from [651].

σ	2	4	8	16	32	64	128
				Average Ranks			
LLPC	2.500	**1.875**	**1.500**	**1.875**	**1.625**	**1.375**	**1.375**
AOC-KK	**2.000**	2.750	3.000	2.875	2.625	2.375	2.000
Invcal	**2.000**	**1.875**	2.375	2.125	2.125	2.275	2.625
Mean Map	3.500	3.500	3.125	3.125	3.625	3.875	-
				Differences, $CD_{<128}$=1.4317, CD_{128}=0.98			
AOC-KK	-0.500	0.875	**1.500**	1.000	1.000	1.000	0.625
Invcal	-0.500	0.000	0.875	0.250	0.500	1.000	**1.250**
Mean Map	1.000	**1.625**	**1.625**	1.250	**2.000**	**2.500**	-



(a) (b)

Fig. 3.11: Average runtime and accuracy of 10-fold cross-validations with LLPC, Invcal, Mean Map, and AOC-KK on several samples of random data. The data was generated for a two Gaussian mixture classification problem ($N = 10000$, $d = 10$, feature values normalized to [0, 1]). ©[2011] Springer. Reprinted, with permission, from [651].

3.3.7.3 Runtime Comparison

For an empirical runtime comparison of the algorithms, we have generated random data for a two Gaussian mixture classification problem (10 000 observations and 10 features, with values normalized to [0, 1]). Then, the average runtime for training and the accuracy of the classifiers for 10-fold cross validation has been assessed for different samples of the data, with varying sizes (see Figure 3.11). The bag size for LLP has been $\sigma = 16$ for all runs. A radial basis kernel with $\gamma = 0.1$ has been used for the kernel methods. LLPC has been run with the exhaustive labeling strategy and fast k-means ($k = 6$), with parameters $maxgen = 3$, $psize = 25$, $mutvar = 1.0$, $crossprob = 0.3$ and $tournsize = 0.25$ for the evolutionary optimization. Both LLPC and AOC-KK used the cluster mean model for prediction.

LLPC shows a high prediction performance for all sample sizes. Moreover, LLPC has the lowest runtime. However, since the methods are implemented in different programming languages (Java, Matlab, R), one should not compare the absolute times, but the slope of the curves. The curve of LLPC's runtime is very flat and almost a straight line, while the slopes of the other curves indicate runtimes that are faster growing. The results empirically demonstrate LLPC's small runtime, which makes the algorithm well suited for resource-constraint settings, especially since centroid cluster models also have a small memory footprint.

3.3.8 Summary and Conclusions

We have presented an approach for LLP known as the Learning from Label Proportions by Clustering (LLPC) algorithm. The approach is general enough to accommodate for the use of different clustering algorithms, labeling strategies, and loss functions. With k-means as the inner clustering algorithm and a constant number of iterations, LLPC has only linear worst-case training time and its cluster mean models are small and fast to apply. In comparison with state-of-the-art methods, which need more training time, the cluster mean models show a significantly higher or equivalent prediction accuracy in the conducted experiments. By training other classifiers on the labeled clusters, the highest achieved accuracy of LLPC is significantly higher for even more bag sizes. Here, LLPC has the highest average rank for all $\sigma > 2$. In addition, LLPC has other beneficial properties of which, to the best of our knowledge, other approaches don't possess all at once: LLPC can handle (1) non-linear decision boundaries, depending on the choice of clustering algorithm, (2) multiple classes, (3) additionally given labeled observations, and (4) it can weight the relevance of features.

LLP has relevance for real-world applications such as guaranteeing the privacy of democratic free elections, or the reconstruction of labels for objects that are hard to track, like those in smart manufacturing. Due to the small memory footprint and fast application of centroid cluster models, as well as a linear training time, LLPC is also well suited for running on resource-constrained devices in the Internet of Things, like edge devices in distributed computing. It has been developed for a series of processing steps in a steel rolling mill allowing for the early quality prediction during the processing (see Section 3.2.4.2). It also has been applied successfully in the field of traffic prediction (see Section 4.1), where it was used to reduce communication costs in a vertically-distributed machine learning setting [655].

Parts of this contribution were previously published in conference proceedings by Springer [651], [654] and in the first author's dissertation [652].

3.4 Simulation and Machine Learning

Petra Wiederkehr
Katharina Morik
Amal Saadallah
Felix Finkeldey

Abstract: The integration of sensor and simulation data for Machine Learning (ML) has become a hot topic. On the one hand, machine learning in the form of Active Learning (AL) allows running exactly those simulations that are needed for predicting future events based on the analysis of explanatory variables. Since simulation is a time-consuming process, we save computational resources by selecting the most informative simulation configurations. On the other hand, simulations represent expert knowledge, so that joining simulation and machine learning from observations leads to better predictions.

The combination of simulation and machine learning has been successfully used to optimize milling processes. Regarding undesirable vibrations of milling tools, a learning-based prediction of a stability criterion is realized. Furthermore, forces of milling operations are predicted using a developed data fusion of sensor and simulation data. Apart from forecasting process characteristics directly, machine learning also identifies parameter values for simulation models. In particular, the machine tool dynamics of a geometric physically based milling simulation system are successfully parametrized for different poses of the machine tool axes to reduce the number of calibration measurements required.

3.4.1 Introduction

In production engineering, many challenges arise regarding process design due to the high complexity and huge variety of engagement situations between the cutting tool and the workpiece to be machined [20]. For milling processes, different process parameter values can lead to different results for the machined component. Especially in the aerospace industry, where deep cavities have to be milled when machining structural components, the required long and slender milling tools can be susceptible to undesirable vibrations that can negatively affect the machined workpiece surface and can lead to increased tool wear. The tool wear can in turn influence the dynamic process behavior [20, 275].

Simulations can support the design of such processes [728]. There are different simulation approaches. Finite element (FE)-based methods represent complex interactions between the cutting edges of the tool and the machined material by numerical

approximations of differential equations [676]. Since the material is constantly being removed from the workpiece, the resulting high degrees of the deformation require relatively small simulation time steps. As a consequence, the simulation runtimes are high and only a few tooth engagements can be simulated in a reasonable time which makes the transfer to complex and long-running processes challenging [208]. By contrast, using geometric physically based simulation approaches, entire processes can be investigated [728]. This is due to the use of simple surrogate models to represent the tool and workpiece. However, this entails that complex interactions, such as tool wear, cannot be modeled reasonably. Moreover, while such methods are significantly faster than FE-based approaches, they are still not real-time capable.

In the context of Industry 4.0, the vision of realizing self-optimizing machining systems by incorporating machine learning methods has emerged recently and started to attract the attention of the manufacturing community [449]. By using these methods, predictions of process characteristics for unseen input data can be achieved in real-time [128, 491]. Several investigations can be found in literature, which focus on the prediction of different process characteristics for milling operations using machine learning methods, e.g. surface roughness [337, 384, 536], process forces [498, 699], or chatter vibrations [482, 690]. Furthermore, over the past decade, fusing multiple sensor signals has been a popular approach to increase the information gain for different tasks, especially for tool condition monitoring [230, 717, 765].

The predictive accuracy of machine learning models can be further improved by combining sensor data with simulation results. Denkena, Dittrich, and Uhlich [170] trained a model using support vector regression. Simulated and measured data are used as features to predict shape deviations of the workpiece. The simulated data is calculated using measured values of the axis positions of the tool. Plakhotnik et al. [503] combined sensor data, such as machine axis positions and spindle torque, with results from computer-aided design and geometry simulations to visualize specific process characteristics and support process design. Peng, Bergs, Schraknepper, Klocke, and Döbbeler [498] utilized FE-based simulations and measurements to train a tool wear-dependent cutting force model based on neural networks.

In this section, different aspects concerning the combination of simulation techniques and machine learning are discussed. As shown in Figure 3.12, it is analyzed how simulations can be replaced by machine learning models to enable real-time predictions [560]. The models learned on simulation data can then be refined by a selected number of experimentally acquired data to close the gap between simulation and experiment, which may be caused by the simplifications of simulation models. In this context, to improve resource efficiency, experiments should only be performed for scenarios whose inclusion is expected to maximize the improvement of the prediction accuracy of the models.

The research concerning AL involves two real-world applications. Reducing the computational resources of an expensive FE simulation is studied in the field of tunnel-

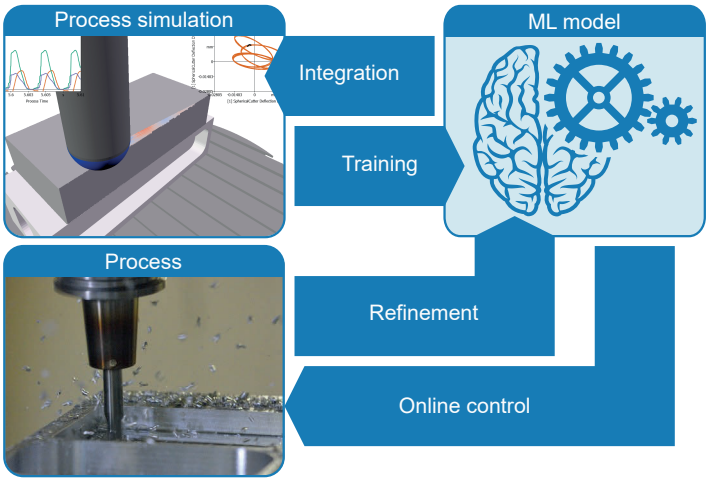


Fig. 3.12: Concept of combining simulation and machine learning [560].

ing (Section 3.4.3.1). The cheaper geometric physical simulation for sampling training data is studied when refining a model for milling (Section 3.4.3.2).

Another focus of the investigation is the fusion of simulation results with sensor data to predict future process characteristics for milling operations [206]. To this end, different fusion strategies are evaluated and compared for the milling application (Section 3.4.4). Furthermore, the first results are obtained for the integration of machine learning methods into a geometric physically based simulation system in order to learn pose-dependent dynamic models [207]. Using ML for initializing simulation models is illustrated by applications in milling (Section 3.4.5.1) and grinding (Section 3.4.5.2).

3.4.2 Simulation of NC-Milling Processes

In milling applications, we use a geometric physically based simulation system [728]. For the tool and workpiece model, the Constructive Solid Geometry (CSG) [210] technique is used. Thereby, the tool model can be realized by modeling the envelope of the rotating tool by combining simple geometric primitives through Boolean operators. For modeling the initial workpiece, the use of a cuboid is usually appropriate. The movement of the tool is defined by discrete positions along an NC path, which can also be used on a real machine. The step size between the tool positions is the feed per tooth defined by the process under consideration. The workpiece geometry of the i-th feed per tooth

$$W_i = W_0 \setminus \bigcup_{j=0}^{i-1} T_j \tag{3.13}$$

is the difference between the model of the initial workpiece W_0 and the union of the tool models $T_j, j = 0, \ldots, i-1$ for all previously processed discrete tool positions along the cutting path. The intersection between W_i and T_i represents the chip shape of the i-th cutting operation. For the calculation of process forces, this chip shape is sampled by rays that have their origin on the rotational axis of the tool. At each ray, the equation

$$F_i = b \cdot k_i \cdot d_0 \cdot \left(\frac{d}{d_0}\right)^{1-m_i},$$
$$d_0 = 1\,\text{mm},\ i \in \{c, n, t\}\ [321] \tag{3.14}$$

is evaluated to calculate process forces in cutting, normal, and tangential directions. Here, d represents the chip thickness, b the width of the cutting segments defined by the rays, and k_c, m_c, k_n, m_n, k_t, and m_t the parameters to be calibrated. Using the directional information of the rays, the force vectors can be transformed into a global coordinate system in x-, y- and z-directions. To simulate tool vibrations, a set of decoupled damped harmonic oscillators represents the dynamic behavior of the machine-spindle-tool system. The parameter values of these oscillators have to be determined using measurements in advance so that deflections in x- and y-directions can be calculated as a vibration-induced displacement of the tool relative to the workpiece [662].

3.4.3 Learning from Simulation

As it was mentioned before, even though geometric physically based simulations allow for the investigation of long-running milling processes in a reasonable runtime [728], they are not yet real-time capable. As a result, predictions can be generated only for process configurations that have been simulated beforehand. By contrast, ML models can be evaluated in real time and, thus, offer the opportunity to predict future unknown events based on an analysis of past data [128] or to classify the current process state, using a set of features extracted from measured data [366]. Therefore, new trends in applied ML aim to replace simulations with ML surrogate models that are more appropriate for real-time applications [116, 212, 559, 560]. In such a manner, not only predictions are generated in real time, but computational resources required for running simulations are also saved.

3.4.3.1 Active Learning for Simulation Data Acquisition

For the simulation of processes that exhibit a high degree of visco-elasticity or elasto-plasticity, numerical simulation methods that require high computational resources are often necessary [116, 212, 559]. Hence, we want to reduce the amount of data that is required to build an accurate surrogate ML model. The same idea was broached in the literature with AL. Starting from a small and non-optimal training set, AL procedures aim at selecting unlabeled data points whose inclusion in the training set improves

the performance of the ML model iteratively [593]. In this context, we develop an AL approach that selects the least amount of simulation configurations to learn an accurate surrogate ML model [559].

One simulation scenario is defined by a given combination of simulation input parameters and considered as a data instance characterized by a set of features. When the simulation runs for a given scenario, it is called a labeled data instance; otherwise, it is called an unlabeled data instance. The active selection can also be decided based on the label proportions of the instances that have so far been generated. In this case, the problem is formulated as Active Class Selection (for more details see Section 5.2.2 in Volume 2).We want to reduce the cost of running process simulations for collecting labeled training data instances. Hence, our Hybrid AL approach (HDAL) combines error-based with distance-based methods to select the minimal number of simulation scenarios that are necessary to build an accurate ML model. We start by randomly selecting a small set of simulation scenarios (i.e. configurations) to run. Then, our framework operates in three stages. First, it trains the ML model on the available labeled scenarios. Second, it computes the training error measures of the ML model for each individual scenario in the labeled set. This step allows us to identify input data regions where the ML model is weak (i.e. uncertain about the label) and probably would need to see more samples from these regions for a better generalization. We select the labeled scenarios with the highest estimated error rates. In the third stage, we determine the closest unlabeled scenarios to these labeled scenarios. The assumption is that two close data instances probably share similar characteristics and thus similar estimated surrogate ML models. However, to avoid clustering problems around these labeled scenarios (i.e. already investigated regions), we devise a "min-max" selection procedure that chooses the furthest ones from the closest unlabeled scenarios. The entire process is iterated until a stopping criterion is met. The stopping criteria for AL procedures is still an open research question. It can be set according to a maximum budget of iterations or when the model accuracy improvement on an independent calibration/validation set over the last iterations becomes insignificant.

We use a 3D FE simulation designed specifically for process-oriented computational simulations of shield tunnelling processes [116] to validate our framework. The simulation models predict soil displacements over time in different measuring points during tunnel excavation given two machine input parameters, namely the grouting and support pressures. Each scenario results in a time series with 64 item steps of displacement observations over 154 monitoring surface points. 20 scenarios are initially selected for training. 130 scenarios are considered unlabeled and 10 are used for testing. The goal is to replace the simulation with an ML model to forecast future soil displacements. To do so, we use Vector Autoregressive with Exogenous time series features (VARX) as a surrogate ML model [559]. For VARX setting, the input data consists of the set of time series of settlements of the surface monitoring points (i.e. measured in millimeters), plus the time series records of the grouting and the support pressures (i.e. measured in *Pa*) of the machines, considered as exogenous variables. A $L1$-regularization is applied

to estimate the model coefficients. The VARX model is retrained with each update of the training set at each AL iteration. Our study shows the ability of our framework in reducing the number of training scenarios required for training up to 60 % while improving the prediction accuracy up to 20 %.

3.4.3.2 Active Learning for Model Building

Active learning can be applied not only to reduce data annotation costs (i.e. reducing the number of simulation configurations to run) but also as an informed sampling procedure to accurately learn the ML surrogate model. The surrogate model should be able to represent the capabilities of the simulation correctly and applied afterward in an online setting to predict process characteristics. Since the quality of ML model depends largely on the training data, we exploit the interaction between the surrogate ML model and the simulation using AL to actively and iteratively sample from the simulation data.

In this context, in order to control milling processes in real time by adapting the process parameter values, we develop a novel ML framework based on a geometric physically based simulation of NC-milling process (cf. Section 3.4.2) [560]. We choose to focus more specifically on building a ML model to predict the Poincaré diameter, which is considered as a process stability criterion for a given simulation scenario, characterized by a given input spindle speed. Parts of this section are already published by the authors [206, 560].

Our experimental use case is a face milling process using a fixed width of cut of 2 mm, an increasing depth of cut from 0 mm to 1 mm, a feed per tooth of 0.1 mm and varying values for the spindle speed between 3000 min^{-1} and 15 000 min^{-1}. For the tool, a torus cutting tool with a diameter of 6 mm and a corner radius of 1 mm is used to machine aluminum alloy 7075.

The proposed framework consists of a weighted ensemble of Multilayer Perceptrons (MLPs) trained on different subsets of features. In addition, an AL procedure is simultaneously used to iteratively design the optimal training set from the generated simulation data. A forecast for the Poincaré diameter is delivered every P_1 milliseconds for the next future P_2 milliseconds. This is achieved by training a committee of MLPs, each on a distinct subset of process features [129]. Each model member of the committee is an online regression model, where the expected Poincaré diameter is assumed to be a function of the historical values of the time series features, namely the forces in the three-dimensional global coordinate system F_x, F_y, and F_z, the deflections in the bi-dimensional space D_x and D_y and the expected values of the chip volume. Since many configurations (i.e. various spindle speed values) are simulated, an AL approach based on the Query By Committee (QBC) paradigm [129] is used to iteratively add simulation configurations, whose inclusion in the training set should improve the prediction accuracy. The final prediction output is computed by a weighted average over the committee members' outputs. The main steps of the framework are illustrated in Figure 3.13.

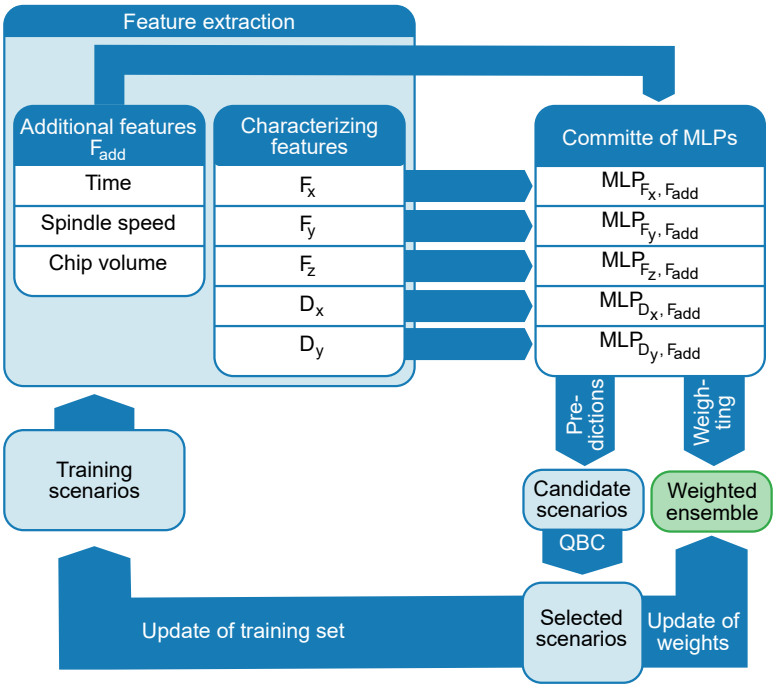


Fig. 3.13: Data acquisition and prediction framework [560].

Pool-Based Active Learning with QBC The output of the simulation depends to a large extent on the input value for the spindle speed. The goal is to identify simulation configurations whose inclusion in the training set improves the accuracy of the surrogate ML model. One way to do this is to carefully design the training set by controlling the selection of training simulation configurations using AL. This control is given by a problem-dependent heuristic, e.g., the decrease of the estimated prediction error on a test set if a given configuration or a pool of configurations are added to the training set. One of the most popular AL approaches is built based on the QBC paradigm [593].

A committee of learners is built following different assumptions on subspaces of instances, which may easily lead to a huge number of hypotheses to cover and quickly become computationally intractable for real applications [442]. It can also be built on disjoint subsets of features [129], usually generated by the Random Subspace Generation (RSG) approach [129], which may also lead to an intractable number of hypotheses. In our approach, each feature subspace represents one characteristic of the original process (cf. Figure 3.13)[560]. The QBC procedure selects the simulation configuration on which the committee members' predictions are maximally split. First, each committee member is trained on an initial small set of simulation configurations N_i. Then, at each iteration, the set of N_c candidate configurations is sorted according

to a disagreement measure for each simulation configuration c_s:

$$f_{QBC}^{c_s} = log\left(\sum_{j=1}^{q}\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t^{c_s} - \hat{y}_{t,j}^{c_s})^2\right), \tag{3.15}$$

where $\hat{y}_{t,j}^{c_s}$ is the predicted value by the committee member j at the instant t, and $\hat{y}_t^{c_s}$ is the estimated result of the committee composed of q models, on the scenario c_s and obtained as follows: $\hat{y}_t^{c_s} = \frac{1}{q}\sum_{j=1}^{q}(\hat{y}_{t,j}^{c_s})$. From the sorted configurations, the first N_s are selected to be added to the training set. This entire process is iterated until a stopping criterion is met, e.g. a maximum number of configurations in the training set is reached, or the accuracy improvement on an independent calibration/validation set over the last iterations becomes insignificant. To output the final predictions, we use the already trained committee of MLPs and aggregate them in a weighted ensemble model. The weights are computed offline using a normalized version of the loss of the model on the training set. During the AL procedure, the weights are updated with each update of the training set. This mechanism may help to achieve a blind adaptation to drifting characteristics in time series observations by adjusting the contribution of each model in the final output [200]. Let $M = \{M_1, M_2, \ldots, M_q\}$ be the committee of q MLPs and $\hat{y}_{t,j}^{c_s}$ is the output of model M_j for a given simulation configuration c_s at a time instant t. The final prediction output is obtained with:

$$\hat{y}_{t+1}^{c_s} = \sum_{j=1}^{q}\frac{\left[\left(1-\chi_j\right)\hat{y}_{t,j}^{c_s}\right]}{\sum_{j=1}^{q}\left(1-\chi_j\right)} : \chi_j \in [0,1], \forall j \tag{3.16}$$

where χ_j is the error of model M_j on the recent obtained training set. To calculate such error, the Normalized Mean Squared Error (NRMSE) is used:

$$NRMSE_{c_s} = \frac{\sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t^{c_s} - y_t^{c_s})^2}{n}}}{\left|\max(Y^{c_s}) - \min(Y^{c_s})\right|}, \tag{3.17}$$

where $Y_s^c = \{y_1^{c_s}, \ldots, y_t^{c_s}, \ldots, y_n^{c_s}\}$ denote a time series of the Poincaré diameter for the simulation scenario c_s and $\hat{y}_t^{c_s}$ is the predicted Poincaré diameter at the given instant t.

Evaluation and Results The simulation generates one observation for each feature with a frequency of 20 kHz, resulting in a step size of 0.05 ms [560]. The simulation configurations are generated for 240 different spindle speed values in the range of 3000 min^{-1} to 15 000 min^{-1} with a step size of 50 min^{-1}. The set of simulation configurations is randomly split into three independent sets. 180 are used for the training, while 20 others served to build a validation set. The remaining 40 are kept for testing. An aggregation period of $P_1 = 10$ ms is used for the preprocessing and a period of $P_2 = 50$ ms is set as a forecasting horizon. The MLP model parameters are tuned using a grid-search procedure on the validation configurations. For the AL procedure, 50 from

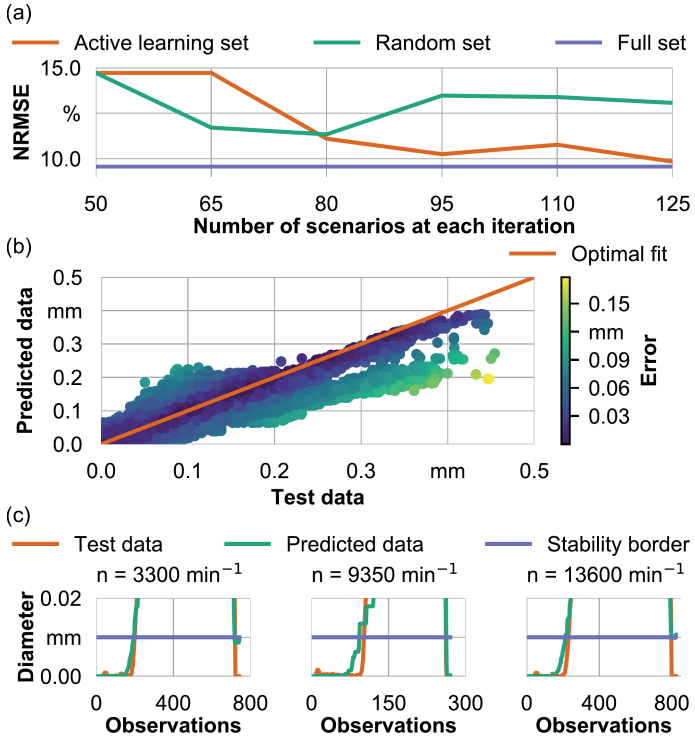


Fig. 3.14: (a) Active learning performance. (b) Comparison between test and predicted data. (c) Depiction of Poincaré predictions in critical areas for three different spindle speeds [560].

the 180 simulation configurations of the training set are randomly selected to construct the initial training set. From the remaining 130 configurations, 15 are added at each iteration. A maximum number of 5 iterations is used as a stopping criterion. The subset of features is constructed using two lagged values of each of the five characteristic features F_x, F_y, F_z, D_x, and D_y. The corresponding spindle speed value, the time index, and two future values of the chip volume are added to enrich each subset of features.

The results are presented from three perspectives [560]: 1) Figure 3.14(a) shows a comparison between the AL approach, a random sampling, and the performance of the ensemble, which is trained using the whole training set; 2) Figure 3.14(b) illustrates a comparison between the predicted Poincaré diameter values and the computed values on a subset of testing scenarios; 3) Figure 3.14(c) presents a comparison between the test and the predicted data for three different spindle speeds.

Figure 3.14(a) illustrates the performance of the AL method. The AL approach reaches almost the same performance as does training over the whole training set, while using only 70 % of the training set. The results show the advantage of designing an

optimal training set especially when a large pool of simulation scenarios with randomly chosen input parameters is available.

Figure 3.14(b) presents a comparison between the true and the forecasted Poincaré diameter values and illustrates that the ensemble is capable of accurately forecasting the Poincaré diameter in distinct scenarios and time periods.

Figure 3.14(c) shows that our framework predicts process instability in time or earlier for three different spindle speeds. This highlights its usefulness for real-time applications, where process parameter values should be monitored and adjusted to avoid process instabilities.

3.4.4 Fusion Between Simulation and Sensor Data

In general, the integration of diverse data sources provides useful and enriched new data. We refer to this combination as a data fusion process [502]. However, it presents many challenges [468]. Fusing different sources can generate conflicts. The conflicts are most often the result of incomplete, erroneous, and out-of-date records [468]. Another challenge comes along with complex sequences that are multi-dimensional, multi-modal and time-varying [331]. In addition, the process of data fusion may also lead to larger amounts of data, which poses difficulties for online application [331]. Solving these challenges requires not only substantial efforts and domain knowledge but also scalable and principled fusion approaches that cope with real-time constraints.

Process simulation can be viewed as background knowledge for domain experts. We want to integrate this knowledge into the machine learning process and, at the same time, use the simulation as an additional data source (i.e. generation of additional data points/data features or annotation of existing data). In this context, we aim at fusing both, simulation and sensor data, to predict active and passive forces in a real-time slot milling process [22, 206]. In [22], we present a framework that allows combining real-world observations collected from sensors and simulations at two levels: the data or the model level. At the data level, observations and synthetic data are integrated to form an enriched dataset for learning. At the model level, the models learned individually from observed and simulated data are integrated using an ensemble technique. Establishing a trade-off between model bias and variance, we perform an automatic selection of the appropriate fusion level. Figure 3.15 shows a schematic illustration of the conducted approach. Parts of this section are already published by the authors [22, 206].

To validate the developed framework, slot milling processes are conducted using a width of cut and depth of cut of 2 mm, a tilt angle of 30° [206]. A ball-end mill with a diameter of 10 mm and two cutting edges is used to machine AISI M3:2, hardened to approx. 62 HRC. During these milling experiments, process forces in x-, y- and z-directions are measured using a triaxial force dynamometer (Kistler 9257B) with a sampling frequency of 20 kHz. The goal of the considered use case is to predict upcoming active and passive forces, which are affected by tool wear. To this end, ten different process

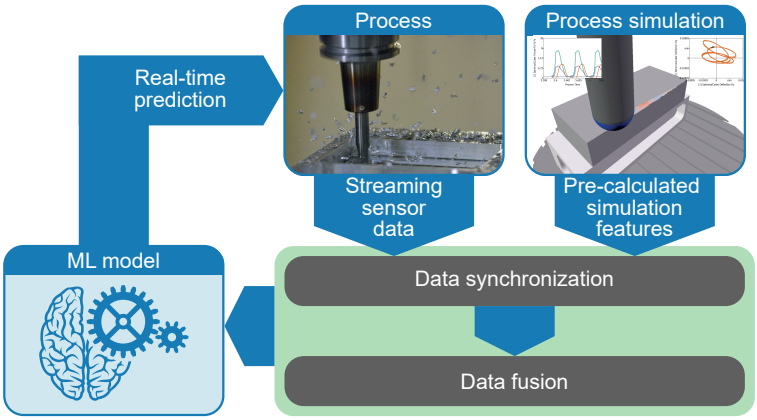


Fig. 3.15: Framework for synchronizing and fusing simulation and sensor data [206].

parameter configurations with varying values for the cutting speed and feed per tooth are used. For each parameter configuration, 180 slots are milled in order to provoke tool wear. After every twentieth machined slot, the width of flank wear is measured for both cutting edges and averaged to obtain a value VB_V which indicates the wear state of the tool [206]. Intermediate values of VB_B for all other slots i are interpolated by

$$\widetilde{VB}_B^i = \begin{cases} \widetilde{VB}_B^{i-1} + \frac{F_\Sigma^i \cdot (VB_B^{j+1} - VB_B^j)}{\sum_k F_\Sigma^k}, & \text{if } F_\Sigma^i > F_\tau \\ \widetilde{VB}_B^{i-1}, & \text{otherwise} \end{cases}, \tag{3.18}$$

$$F_\Sigma^i = |F_x^i| + |F_y^i| + |F_z^i|, \tag{3.19}$$

where $i \neq 0$ and F_τ is a threshold that has to be defined in order to distinguish between signal and noise. The values VB_B^j and VB_B^{j+1} are estimated by measurements between which the interpolation is performed and k is the number of interpolated values between these measured values. Using this approach, high force amplitudes are assumed to induce a high load on the cutting edges, resulting in increased values for VB_B [206].

3.4.4.1 Simulation-Sensor Data Mismatch Evaluation

The fusion of data collected from both sources into a single data representation is not straightforward and data mismatch between both sources needs to be checked. Data mismatch may mean that different data sources attribute different values to the same instance. This can be addressed by ensuring the *completeness* and the *correctness* of each data source [468].

The mismatch is often caused by different data alignments due to different data sampling frequencies from simulations and measurements. This results in a time delay between simulated and measured data. Therefore, data synchronization is required.

In addition, the mismatch can be caused by learning from different underlying distributions. In fact, due to the simplified models in the simulation system and the negligence of complex engagement behaviors, e.g. frictional effects, which are used to ensure a reasonable runtime, non-negligible deviations between simulation data and measurements of the corresponding process characteristics can occur, especially for process forces or tool vibrations. This deviation can differ for different process parameter values. As a result, a calibration of the simulation is required. Due to measurement noise and uncertainties, the quantification of simulation accuracy is a challenging task.

Synchronization If both, simulated and sensor data, are acquired using the same sampling frequency, only a constant time shift between each time series has to be determined. This can be achieved, e.g., manually, using change points, estimated by auto-regressive approaches [740] or by analyzing the continuous wavelet transform of the time series. In the context of the latter approach, the transformed signal is given by:

$$W(a,b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} \Xi(t) \Psi^{\star} \left(\frac{t-b}{a} \right) dt, \tag{3.20}$$

where $\Xi(t)$ is the original signal, $\Psi^{\star}(\frac{t-b}{a})$ represents the complex conjugate of a scaled and translated mother wavelet $\Psi(t)$, and a and b are the scale and translation parameters, respectively. Each scale corresponds to a frequency, resulting in information about the correlation between a given signal at a certain time instant and an investigated frequency without the need to make a trade-off between time and frequency resolution, which is a crucial issue of spectral analysis. In the milling application, the time-related delay between the two investigated time series can be identified by the points in time where the intensity of the wavelet transform at the tooth engagement frequency is greater than zero.

Calibration For the considered geometric physically based simulation system, simulation models can be calibrated using measurements as ground truth. This has to be performed for each combination of the tool geometry and the workpiece material of the regarded process. For simulated forces, for example, the parameters of the force model ρ can be determined by applying an optimization procedure to minimize the squared Euclidean distance between simulated and measured forces

$$\ell(\rho) = \sum_{i=1}^{n} \left(\boldsymbol{F}_{\text{sen}}(t_i) - \boldsymbol{F}_{\text{sim}}(\rho, t_i) \right)^2, \tag{3.21}$$

acquired using the same process parameter values for both, the machining process and the corresponding simulation conduction. To this end, any optimization algorithm could be applied to solve the minimization task. However, in practice, quasi-Newton approaches often outperform other methods. Using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [113] optimization algorithm, for example, an approximation of the

Hessian matrix H_k is estimated, which is updated at each iteration k of the procedure. According to Newton's method, the parameter values of the next iteration

$$\rho_{k+1} = \rho_k + \alpha_k s_k \tag{3.22}$$

are given by the line search along the descent direction

$$s_k = -H_k \nabla \mathcal{L}(\rho_k) \tag{3.23}$$

by estimating α_k through

$$\hat{\alpha}_k = \underset{\alpha}{\operatorname{argmin}}\, \mathcal{L}(\rho_k + \alpha s_k). \tag{3.24}$$

The update of H_k is performed by adding a rank-two correction

$$H_{k+1} = H_k + auu^T + bvv^T, \tag{3.25}$$

$$u = \delta_k = \rho_{k+1} - \rho_k, v = H_k \gamma_k = H_k \nabla \mathcal{L}(\rho_{k+1}) - \nabla \mathcal{L}(\rho_k) \tag{3.26}$$

are typically chosen, so that the quasi-Newton condition

$$H_{k+1}\gamma_k = H_k \gamma_k + auu^T \gamma_k + bvv^T \gamma_k = \delta_k \tag{3.27}$$

is satisfied, resulting in

$$H_{k+1} = H_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{H_k \gamma_k \gamma_k^T H_k}{\gamma_k^T H_k \gamma_k}. \tag{3.28}$$

3.4.4.2 Automatic Fusion-Level Selection

A sophisticated ML model should establish a trade-off between bias and variance. Such statement gives a guidance on how to automate the decision for the fusion-level selection. In fact, given a learned model $\hat{f}$ that approximates an unknown true model f, the expected mean-squared error between the target variable $y = f(x)$ and the model predictions on an unseen sample x, can be decomposed into bias, variance, and an irreducible error term [334]. One way to reduce the variance-type error is to use an ensemble model [101], that combines many models into one single model using an averaging technique [556]. Such a statement is derived from the ensemble error decomposition into the average bias of the ensemble single models, variance, and a covariance term [694]. Brown, Wyatt, and Tino [106] have proven that when using an average-based ensemble model $\bar{f}$ with equal weights (i.e. $\bar{f} = \sum_{i=1}^{N} w_i f_i$, $f_i, i \in \{1, \cdots, N\}$ are the single models), the expected error decomposition is given by:

$$\mathbb{E}\left[\left(f(x) - \bar{f}(x)\right)^2\right] = \overline{Bias}^2 + \frac{1}{N}\overline{Var} + (1 - \frac{1}{N})\overline{Covar}, \tag{3.29}$$

where

$$\overline{Var} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\left(\hat{f}_i(x) - \mathbb{E}\left[\hat{f}_i(x)\right]\right)^2\right], \tag{3.30}$$

$$\overline{Bias} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbb{E}\left[\hat{f}_i(x)\right] - \mathbb{E}\left[f(x)\right]\right), \tag{3.31}$$

$$\overline{Covar} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1,\neq i}^{N}\mathbb{E}\left[\left(\hat{f}_i(x) - \mathbb{E}\left[\hat{f}_i(x)\right]\right)\left(\hat{f}_j(x) - \mathbb{E}\left[\hat{f}_j(x)\right]\right)\right]. \tag{3.32}$$

The variance in Equation 3.29 is the average variance divided by the number of base models N. When N is big enough, the variance term in Equation 3.29 will diminish. However, Equation 3.29 states that also the averaged bias and covariance should be taken into account while adding more and more models. In our setting, we are concerned with a small number of base models (i.e. mainly 2, one built on sensor data and the other one on simulation data). In addition, the decision of the transition from a single model to an ensemble has to be made. Therefore, it is more straightforward to deal with the whole term $\frac{1}{N}\overline{Var} + (1 - \frac{1}{N})\overline{Covar}$, as a variance-type error for the ensemble model and the corresponding bias as the average bias of single models $\overline{Bias}$, since

$$Bias(\bar{f}) = \left(\mathbb{E}\left[\bar{f}\right] - f\right) = \left(\mathbb{E}\left[\sum_{i=1}^{N}\frac{1}{N}f_i\right] - f\right) = \overline{Bias}. \tag{3.33}$$

The decision for the transition from a *data-based* fusion to a *model-based* fusion should be based on the level of the expected variance-type error together with the expected bias of the *data-based* fusion model. Let $f_{simulation}$ and $f_{sensors}$, two models each trained on simulation and sensors data, respectively, and a model f_{fus} be trained using a *data-based* fusion approach. From the decomposition in Equation 3.29, we can conclude that the *model-based* fusion using an averaged ensemble with equal weights contribute to reducing the variance-type error compared to the *data-based* fusion model if

$$Covar(f_{\text{simulation}}, f_{\text{sensors}}) \leq \tau, \tag{3.34}$$

where the threshold $\tau = 2\left(Var(f_{\text{fus}}) - \frac{Var(f_{\text{simulation}}) + Var(f_{\text{sensors}})}{2}\right)$.

Equation 3.34 complies with the decomposition in Equation 3.29, stating that a lower covariance is always desired to reduce the overall ensemble error. It also confirms that enforcing a degree of diversity between the ensemble members through low covariance is favorable as it reduces the ensemble ambiguity, presented in the more general ensemble error decomposition schema [355]. Furthermore, once the single models are trained and built, the average covariance term can be estimated entirely without any knowledge of the true data labels or the real function f to be approximated. From a practical point of view, this result confirms the usefulness of using simulation for enriching data with samples that reflect different patterns than the ones observed with

sensor data. This enrichment includes either new features that cannot be measured by sensors or observations that cannot be detected with sensors.

The bias of the ensemble model will be equal to the average bias of the single models in case of equal weights. So, it will achieve a lower bias than the single model with the highest bias. If the covariance between single models is lower than the threshold τ in Equation 3.34, then the system compares their average bias with the bias of the *data-based* fusion model to check if the variance-type error reduction with the average bias will establish a better bias-variance trade-off or not. If it is not, the system sticks to the *data-level* fusion.

The systematic fusion-level selection is validated by the previously described slot milling process [206]. We measure the process forces (F_a and F_p) using sensors, aggregate them using an aggregation period of 0.1 ms, and predict the future expected forces each 0.1 ms for the next 10 ms. Monitoring the milling process forces enables control of both, process stability and quality [560]. 10 different process scenarios are investigated by varying the input parameters, namely speed, and feed. The resulting length of the time series for each scenario depended on the values of the input parameters and varied from 13 250 to 54 500 observations. The simulation is used for feature enrichment by generating features that typically cannot be measured during the process. For this purpose, the chip volume, the sum of time of engagement, the feed, and the mean of the cutting speeds are generated for each point in time to potentially enrich the feature space of the force measurements [206].

After solving possible mismatches between the sources, a unified feature set is created by joining new features generated by the simulation together with the lagged sensor measurements of the forces as sensor features [206]. The Random Forest regressor (RF) [101] is chosen for the prediction of F_a and F_p. The results are presented for 10 cross-validation folds, where 9 scenarios (i.e. time series) are kept for training and 1 scenario for testing for each fold [206]. The prediction error is evaluated using the NRMSE (Equation 3.17). We used 5 lagged values for the time series of the forces as sensor features. For each time step, lagged sensor values are joined together with the simulation features for the current time step (i.e. simulation features are pre-calculated and stored, and only sensor data is streaming). We have also devised a binary feature based on the simulation features called the *activity feature*, which indicates the engagement situation of the tool (0: no engagement, 1: engagement) and is added to the fused set of features.

The results in Table 3.5 show that the feature-based fusion model outperformed models trained separately on each data source. The feature-based fusion is automatically selected as the best way to perform data fusion using simulation and sensor data after empirically computing the threshold τ derived in Equation 3.34 for the covariance and the average bias of the single models trained separately on each data source. Our theoretical insights are validated by showing a comparison with the model-based fusion [206]. These results are presented in Table 3.5. Furthermore, examples of empirical evaluations of the covariance, the threshold τ derived in Equation 3.34, the average

Tab. 3.5: Comparison between the NRMSE of predicted active and passive forces using different methods.

Method	$RF_{Simulation}$	$RF_{Sensors}$	$RF_{Feature\text{-}based\ fusion}$	$RF_{Model\text{-}based\ fusion}$
NRMSE F_a	15.29 % ± 3.70 %	24.24 % ± 5.95 %	9.95 % ± 2.90 %	16.25 % ± 4.27 %
NRMSE F_p	21.50 % ± 11.15 %	33.63 % ± 5.38 %	13.30 % ± 6.30 %	19.69 % ± 5.37 %

Tab. 3.6: Comparison between different measures for the fusion-level selection.

Measure	F_a	F_p
Covar ($RF_{Simulation}$, $RF_{Sensors}$)	21.18	4750.80
Threshold τ	109.92	-13092.95
Average bias ($RF_{Simulation}$, $RF_{Sensors}$)	782.50	13714.16
Bias ($RF_{Feature\text{-}based\ fusion}$)	121.21	3204.13
Var ($RF_{Feature\text{-}based\ fusion}$)	330.40	8614.31
Var ($RF_{Model\text{-}based\ fusion}$)	140.86	13367.52

bias of single models, and the empirical bias of the feature-based fusion model for the predictions of F_a and F_p are shown in Table 3.6 to describe how the fusion level selection is made. In addition, the model variances of the model-based (i.e. ensemble) fusion and the feature-based fusion are reported. For F_a, the value of the covariance between single models is lower than the threshold, which guarantees that computing the ensemble model will reduce the variance type error and this is also confirmed by the reported empirical variance values in Table 3.6. However, validating the covariance threshold is not sufficient. The average bias of single models should also be compared to the bias of the feature-based fusion model. Comparing these values clarifies whether the model-based fusion will contribute to reducing the variance type error, but also alters the bias by increasing it with approximately a factor of six. For F_p, the value of the covariance between single models is higher than the threshold, which indicates that computing the ensemble model will not contribute to reducing the variance type error. In addition, the bias of the feature-based fusion is lower than the reported average bias. This observation confirms that the model-based fusion will reduce neither the variance nor the bias.

3.4.5 Initialization of Simulation Models Using ML Methods

First investigations are performed with respect to the initialization of simulation models using machine learning methods. To this end, two different simulation systems are considered, which are developed to investigate the milling and grinding processes, respectively. Parts of this section have already been published by the authors [207, 727].

3.4.5.1 Learning of Frequency Response Functions and Compliance Models

The underlying system of a geometric physically based milling simulation (cf. Section 3.4.2) models tool vibrations by a set of decoupled damped harmonic oscillators [662]. The parameters of these oscillators have to be calibrated by measurements of the Frequency Response Function (FRF) of the considered machine-spindle-tool system, acquired by impact hammer tests [325]. In addition, the dynamic behavior of this system changes with different poses of the tool. The modeling is performed separately for all considered spatial directions- Here this corresponded to the x- and y- directions. Each oscillator is parameterized by identifying values for the modal mass m_m, the natural frequency f_m, and the damping constant γ_m. The complex response function [207]

$$\begin{aligned} G(\omega j)_d &= \sum_{q=1}^{Q} \left(\cos(\phi_d^{(q)}(\omega)) \cdot A_d^{(q)}(\omega) + \sin(\phi_d^{(q)}(\omega)) \cdot A_d^{(q)}(\omega) \cdot j \right), \\ d &\in \{x, y\}, \\ A(\omega) &= \frac{\frac{1}{m_\mathrm{m}}}{\sqrt{\left(\omega_\mathrm{m}^2 - \omega^2\right)^2 + 4 \cdot \gamma_\mathrm{m}^2 \omega_\mathrm{m}^2}}, \\ \phi(\omega) &= \tan^{-1}\left(\frac{2\gamma_\mathrm{m}\omega}{\omega_\mathrm{m}^2 - \omega^2} \right), \\ \omega_\mathrm{m} &= 2\pi f_\mathrm{m} \end{aligned} \tag{3.35}$$

is represented by superposing the amplitude and phase of the parameterized oscillators q for each angular frequency ω. By using measured FRFs, the loss function

$$\ell = \sum_i \left(\tilde{A_t}^{(i)} - \tilde{A_p}^{(i)} \right)^2 + \sum_i \left(\tilde{\phi_t}^{(i)} - \tilde{\phi_p}^{(i)} \right)^2, \tag{3.36}$$

can be derived, which is the sum of the squared deviations between the normalized values of the calculated amplitudes $\tilde{A_p}$ and phase $\tilde{\phi_p}$ and the normalized measured data $\tilde{A_t}$ and $\tilde{\phi_t}$.

Two different learning tasks are pursued in the performed research. On the one hand, pose-dependent FRFs are learned in order to reduce the measurement effort. On the other hand, the resource- and time-consuming and only semi-automatic task of calibrating the oscillator parameter values for different tool poses is replaced by machine learning.

The amount of data needed to perform the investigations is obtained by frequency response measurements in our laboratory for two different machine tools, Heller FT 4000 (M_1) and DMG HSC 75 linear (M_2), using a centrally composed statistical experimental design with star points. A total of 46 and 49 poses are measured for M_1 and M_2, respectively. An impulse hammer (Kister 8206) are used to excite a ball end mill (Fraisa X7400) with a diameter of 10 mm. The impulse response is measured by an accelerometer (PCB Piezotronics 352C23) attached to the tool tip [207].

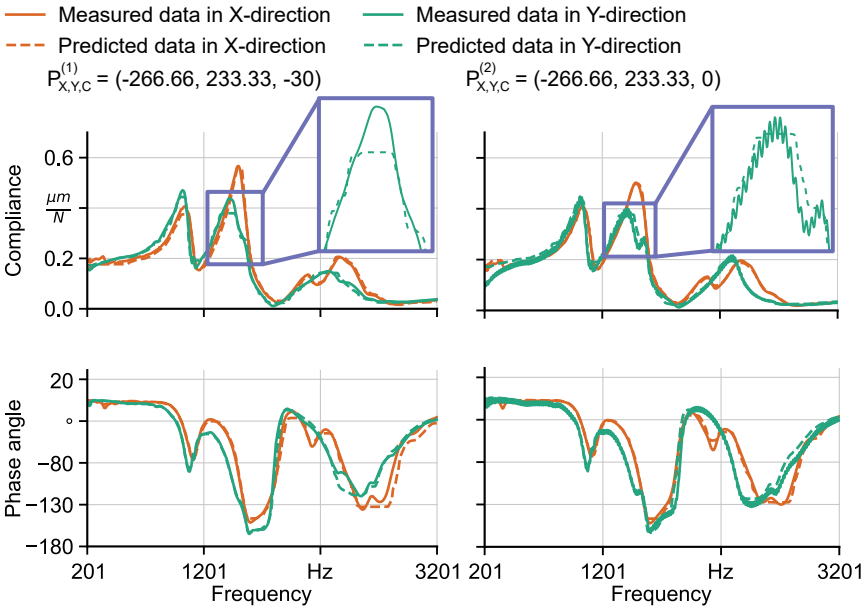


Fig. 3.16: Predicted and measured FRFs for machine tool M_1 [207].

As mentioned, two different objectives are investigated. For both objectives, let $\mathcal{X}$ be a set of $J \times N$ features sampled from an unknown distribution $\mathcal{D}$ and $\mathcal{Y}$ be a set of $K \times N$ targets labeled by a labeling function. The first objective involved the prediction of FRFs. For this, the measured FRFs of all P considered measurement poses are discretized into data points by the frequency resolution Δf such that M is the number of frequencies examined for each pose. Each of the $N = P \cdot M$ data points contains a number of K targets that included compliance amplitude and phase shift for the x- and y-directions of the machine coordinate system. Let J be the number of features consisting of the frequency and the positions of the three axes defining the pose. For the second learning task, which is the prediction of modal parameter values for given poses, let J consist of the three pose-dependent features and $N = P$. For the targets, let $K = 3 \cdot (Q_x + Q_y)$, where Q_x and Q_y are the number of oscillators in the x- and y-directions, respectively. Using this approach, the learning task attempts to represent the relationship between different interdependent oscillators of each compliance model across the two different vibration directions as well. For both learning tasks, the goal is to find a learner $h : \mathcal{X} \to \mathcal{Y}$ with respect to the distribution $\mathcal{D}$ [207].

Figure 3.16 shows an exemplary comparison between measured and predicted FRFs for two different poses using machine tool M_1. The phases are predicted with high accuracy in both x- and y-directions for both test poses. The amplitudes in the x-direction are predicted with a nearly non-visible deviation from the measured curves for both tests poses. There are two peaks visible in measured FRFs in y-the direction

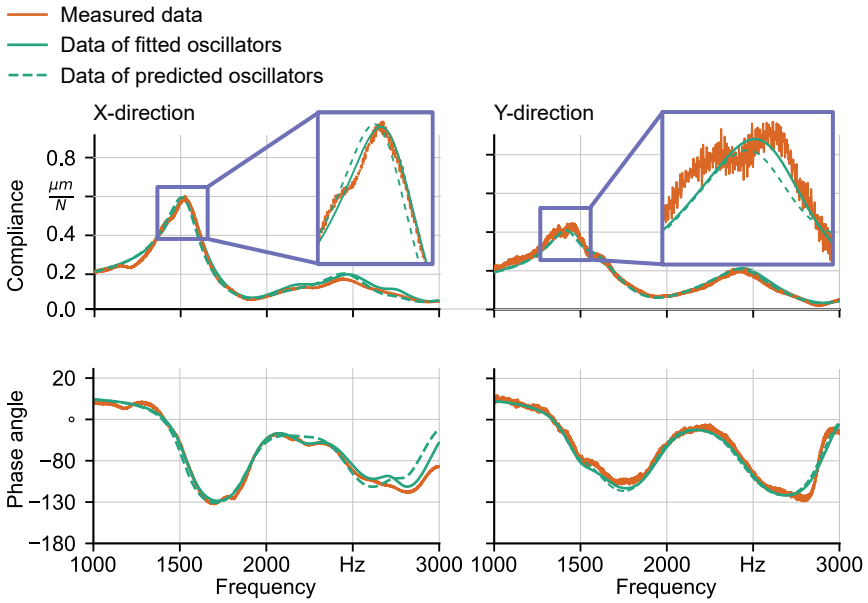


Fig. 3.17: FRFs resulting from predicted and manually calibrated compliance models for machine tool M_2 [207].

in a frequency range of 1200 Hz to 1700 Hz, which are very distinct for $P^{(2)}_{\mathrm{X,Y,Z}}$. For $P^{(1)}_{\mathrm{X,Y,Z}}$ the second peak can hardly be detected. This behavior could be represented by the model to a certain extent. In addition, the model also predicts a visible second peak for the remaining test poses, but the differences in the distinction between the two peaks across the poses can not be achieved. This effect is observed to be minimal in the data examined and needs to be analyzed in more detail in future research activities. In order to represent such behavior, more observations in which the fusion of peaks is present and that can be considered for the training procedure would be necessary.

Figure 3.17 shows a comparison between FRFs that resulted from predicted and manually calibrated compliance models in x- and y-direction for one specific pose using machine tool M_2. Generally, a high accordance is observed. Examining the zoomed-in areas of the FRFs, it can be seen that the shape of the measured FRF can only be represented coarsely by the manual fitting procedure. Since the fitted oscillator parameter values serve as target values for the learning task, the FRF, which is calculated based on predicted oscillator parameter values, can not reproduce the measured behavior in higher detail than the FRF that results from the fitting procedure. Nevertheless, there are only small deviations between the fitted and predicted data. Therefore, the learning of oscillator parameter values directly from given poses can be interpreted as successful.

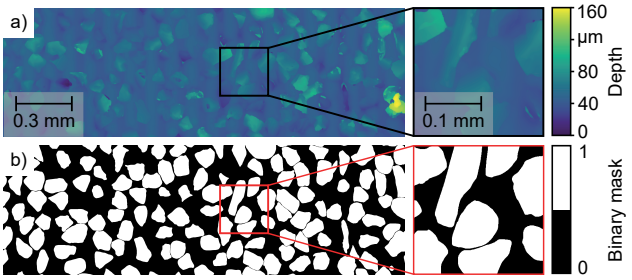


Fig. 3.18: Images of (a) measured depth information of grinding tool topographies and (b) the corresponding segmentation mask [727].

3.4.5.2 Augmented Semantic Segmentation for the Digitization of Grinding Tools

The following results are based on the investigations of [727]. The stochastic nature of grinding processes entails several challenges regarding process simulations. For modeling grinding tools, the choice of the methods for representing the individual grains and grain shapes have a significant influence on the accuracy of simulation results. Especially for single-grain scratch simulations and FE-based analyses, the identification of grains that adequately represent the overall characteristics of the tool used is crucial. In order to perform this identification successfully, an analysis of a huge amount of grains that have to be manually separated from the bond, is necessary. We developed a learning-based methodology, to automate this separation for digitized grinding tools by semantic segmentation. We have focused in particular on evaluating the prediction accuracy of the grain boundaries to be able to distinguish neighboring grains. This is crucial for a subsequent automated extraction of the grains. Figure 3.18a visualizes the measured depth information of grains in the bond. In addition, a manually generated segmentation mask is shown in Figure 3.18b. For the semantic segmentation, a novel neural network architecture [727] is developed, which is based on Fully Convolutional Networks (FCN) [420] (see Figure 3.19). In contrast to conventional FCN architectures found in literature, the channel information is gradually down-sampled by half in each transposed convolution layer instead of performing a single reduction operation. In addition, the up-sampling of the image dimensions is also spread across three twofold up-sampling steps.

Out of 4678 grains, 500 are used for testing purposes [727]. For hyper-parameter identification, random search [57] is used. In order to evaluate the prediction accuracy, the pixel accuracy [420] PACC $= \sum_i N_{ii} / \sum_i M_i$ is used, where N_{ij} is the number of pixels of a class i which are predicted to belong to class j and $M_i = \sum_j N_{ij}$ is the number of pixels of class i. In addition, the boundary pixel accuracy (BPACC) is calculated as the pixel accuracy of the boundary pixels of each grain, which are estimated using a border following algorithm [665].

Figure 3.20 shows the results for grain segmentation for different numbers of grains incorporated for training using the developed approach versus applying an FCN-8. The

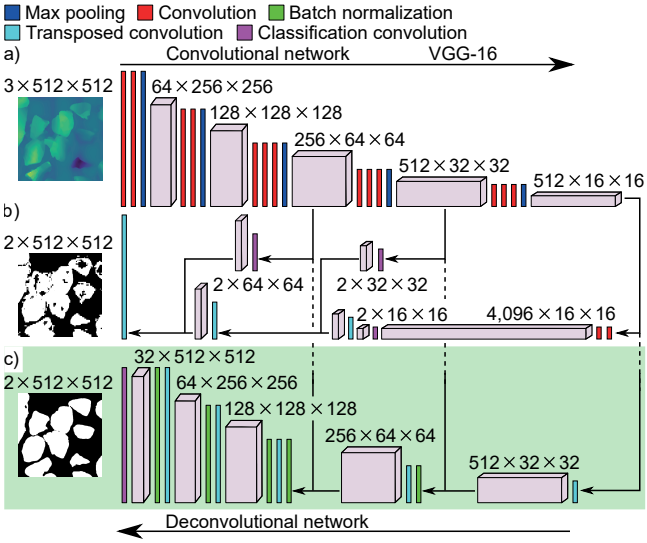


Fig. 3.19: FCN architecture including (a) the convolutional network, (b) a deconvolutional network based on FCN-8, and (c) the developed approach for the deconvolutional network [727].

conventional FCN-8 delivers insufficient results for all three investigated numbers of grains used for training. By contrast, using the developed FCN architecture, it is even possible to distinguish closely neighboring grains, if 2575 or 4178 grains are used for training [727].

To successfully train ML models, manual segmentation still has to be conducted in order to establish the required feature/target correspondences. To this end, data augmentation is used, to drastically reduce the necessary number of measurements. Different image manipulation techniques [605], e.g., rotation, flipping, or noise injection, are combined in a random sequence, to increase the amount of training data. In order to quantify the degree of augmentation, the augmentation factor per image (AFPI) is used as the number of generated images for each image based on measurements in a combined training set. Figure 3.21 shows the segmentation results using different numbers of grains used for training and different values for the AFPI, for the PACC, and the BPACC. The PACC value is higher the more grains are used and the higher the AFPI is. However, for the BPACC, a local optimum can be identified, indicating that high PACC values will not necessarily result in good segmentation results of the grain boundaries. Furthermore, choosing the AFPI as low as possible results in significantly lower training runtime [727]. Since grinding tools are constantly affected by tool wear during the process, the transferability of the model corresponding to the local optimum to different states of tool wear is also investigated. For further details, see the corresponding publication [727].

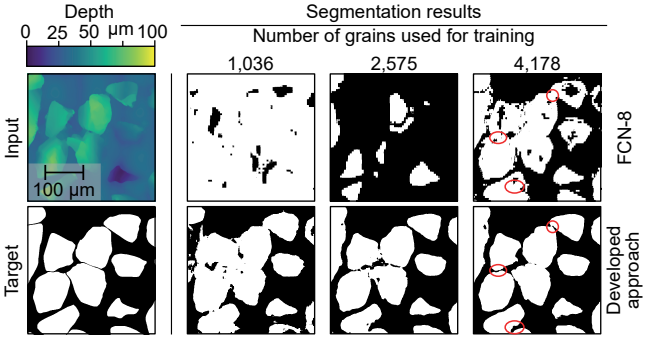


Fig. 3.20: Segmentation results using the developed approach in comparison with using an FCN-8 [727].

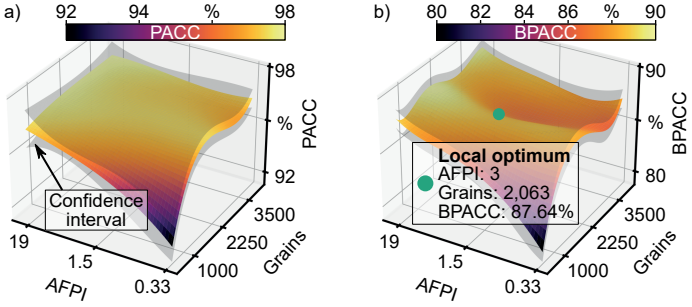


Fig. 3.21: Segmentation results with varying number of grains used for training and values of the AFPI [727].

3.4.6 Summary and Conclusions

This contribution has presented investigations that highlight different aspects of the combination of sensor and simulation data for scientific insights into machining. The methods are illustrated by milling, grinding, and tunneling processes as case studies. The main resource restriction in machining applications is processing time. Ensemble learning methods enable the real-time capability of simulated predictions. ML enhances the simulation. Another resource, which is often restricted in real-world applications, is (labeled) data. Various data fusion strategies were discussed to combine sensor and simulation data for real-time predictions of process characteristics of milling operations. Here, simulations and ML help each other.

In addition, ML methods were used to initialize simulation models, specifically the dynamic model of a geometric physics-based milling simulation system and the tool model of a grinding simulation. ML helps the simulation.

The presented results emphasize the significant potential of combining sensor data, simulation results, and ML methods for the analysis and optimization of manufacturing processes in the context of Industry 4.0.

3.5 High-Precision Wireless Localization

Janis Tiemann

Abstract: Recent developments in Ultra-Wideband (UWB) wireless communication enable wireless localization as a link between the digital and the physical world. With the technological advances, the achievable precision and accuracy is increased dramatically such that novel applications exploiting this precise cyber-physical link become feasible. Autonomous swarms of robots, precise and scalable tracking of goods or safety applications are within the reach of this potential. However, the increase in communication required for such capabilities to become feasible is constrained by bounds of channel utilization, energy consumption, and intelligent information distribution. Therefore, novel approaches for maximizing information and localization throughput while minimizing channel utilization and power consumption and maintaining precise localization results are crucial to overcoming technology barriers and ultimately enabling a connected cyber-physical world. Promising approaches are novel localization-specific protocols to coordinate channel access among localization targets in order to achieve reliable data rates while minimizing actual power consumption. Further, intelligent approaches are required to increase the achievable accuracy for these resource-efficient localization approaches such as Time-Difference of Arrival (TDOA). In those cases, additional parameters of the radio channel can be exploited to obtain quality indicators for measurement and mitigate outliers or generally improve the localization accuracy through adequate estimation. In the following, the requirements of several applications are analyzed, and an overview of the solution space in terms of channel utilization, energy efficiency, and accuracy is given. Based on these requirements, solution approaches are presented to improve both channel utilization and energy efficiency. Further, approaches to increase the achievable accuracy in challenging environments are illustrated and evaluated. It is shown that novel approaches for high-precision wireless localization enable novel applications by employing localization-specific protocols and methods to improve the accuracy despite challenging conditions.

3.5.1 Introduction: Precise, yet Scalable Wireless Localization

Wireless localization is seen as an enabler for many applications requiring a link between the physical and the digital world. Many approaches exist to achieve this connection utilizing a wide range of technologies. An overview of the capabilities in terms of accuracy and range of those technologies is given in Figure 3.22. The diagram illustrates the difference between widely adapted communication technologies, such as cellu-

lar networks and consumer-grade wireless standards, and more application-targeted standards, such as UWB. Early methods for localization were solely based on cell-id and/or sector differentiation. However, novel approaches are capable of increasing the accuracy for tracking and guiding applications. More dedicated localization systems like Global Navigation Satellite Systems (GNSS) can achieve exact localization results but are limited to outdoor Line of Sight (LOS) operations, as they suffer severely from multi-path fading. Here, UWB technology is key to overcoming the technology barrier, enabling highly precise indoor localization through precise Time Of Arrival (TOA) estimation. For localization methods enabled by 5G mmWave communications, see Section 5.5.

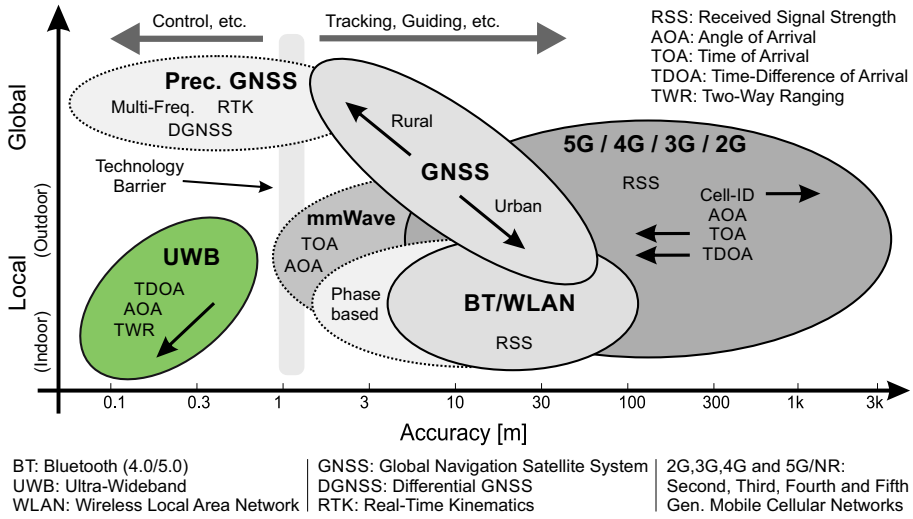


Fig. 3.22: Illustration of the field of localization capabilities of wireless communication technologies in terms of accuracy, range, and place of usage.

Hence, a significant amount of research is challenged by this newly available technology. Integrated UWB solutions enabled a new degree of accuracy in wireless localization. In contrast to many signal strength-based methods, these TOA-based approaches emerged as the most promising candidate for accurate and reliable measurements in the centimeter range. Due to the usage of high bandwidths that enable sharp pulse-based modulation, these UWB systems are capable of resolving many multi path-induced errors in TOA estimation [228].

Due to this reason, research in several areas of application arose utilizing this newly available connection between the digital and the analog world. One particular area of interest is the field of massively scalable and low-power localization for logistics

in the Industry 4.0. Here, low-power localization devices are key to monitoring and optimizing the whole production process, as illustrated in Figure 3.23.

For large-scale deployment, however, the scope of the most current research is insufficient as it neglects scalability and multi-user interference by utilizing localization approaches that require the exchange of many messages. This introduces not only significant resource usage in terms of channel utilization, but also requires substantial usage of energy for message exchanges. In summary, the addressed points of this section are listed as:

- **Massive Multi-User Scalability**
- **Minimal Energy Consumption at the Mobile Units**
- **High Accuracy Suitable for Control-Grade Applications**

3.5.2 Related Work: Evolution of wireless localization within CRC

The methodological continuity of the work within the CRC is illustrated by several publications:

- **UWB Indoor Positioning for UAVs (Unmanned Aerial Vehicles)** [682] uses Two-Way Ranging to enable UAV Indoor Navigation. Limitations in multi-user scalability motivated further research.
- **Multi-User Interference Analysis** [685] is a detailed look into multi-user interference for UWB systems and wireless clock synchronization. Further analysis is conducted in [687] which provides an analytical model for the interference.
- **ATLAS - TDOA-Based Localization** [684] overcomes multi-user scalability limitations and presents an open-source approach.
- **Scalable Multi-UAV Indoor Navigation** [688] demonstrates the scalability and accuracy for control-grade systems.

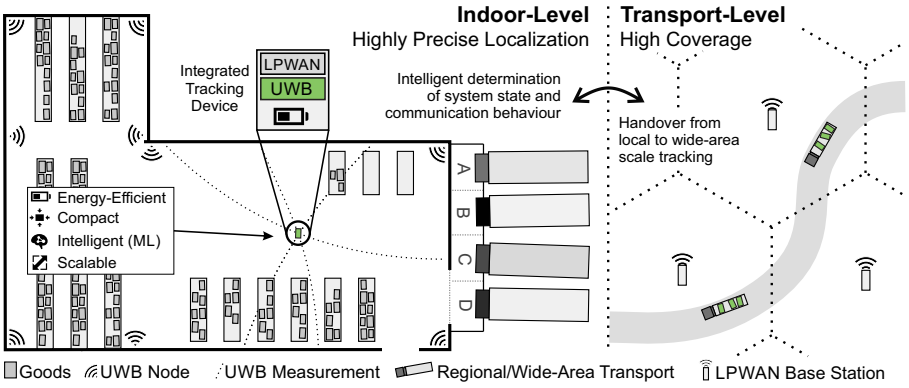


Fig. 3.23: Illustration of a potential usage scenario.

- **Enhanced UAV Indoor Navigation** [681] improves the precision and range for control-grade systems.
- Further work provides an open-source extension **ATLAS FaST** to the ATLAS approach in order to improve energy efficiency and reliability [678, 679, 687].
- Finally, the PhD thesis **Scalability, Real-Time Capabilities and Energy Efficiency in Ultra-Wideband Localization** incorporates and extends the key findings of the research, see [683].

3.5.3 Approaches: Scalable, Real-Time Capable Energy Efficient Localization through UWB

The following sections present approaches for high precision wireless localization based on the work in [680, 683, 687]. The approaches are tailored to find a sweet spot in the trade-off between multi-user scalability, energy efficiency, and achievable accuracy for wireless localization. In a first step, the underlying ATLAS localization system is highlighted briefly.

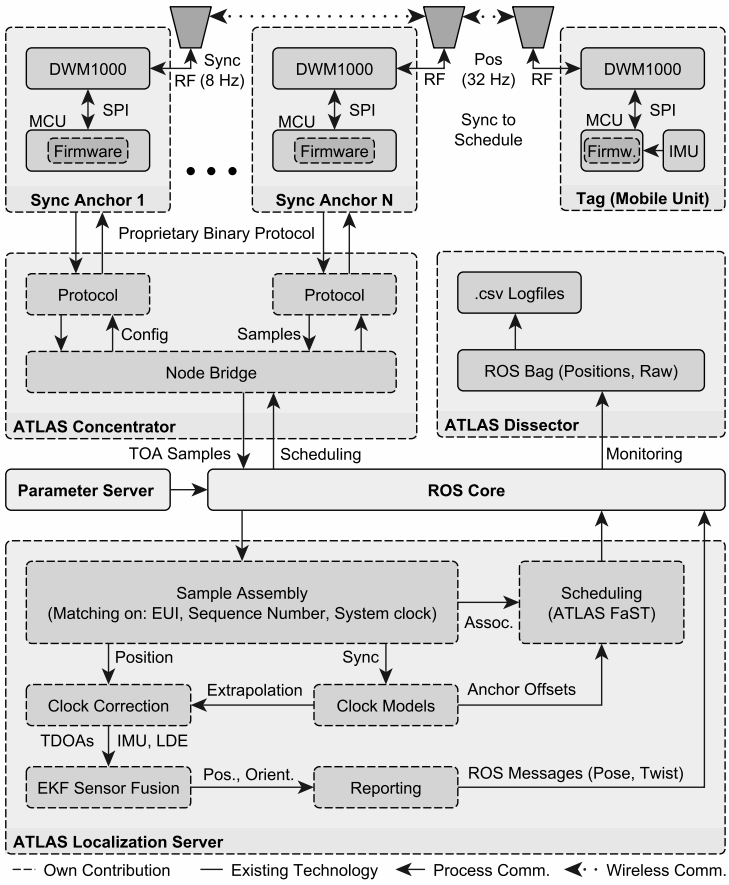


Fig. 3.24: System architecture of the ATLAS RTLS utilizing the FaST scheduling approach. The approach is based on the robot operating system (ROS) to allow for flexible and modular design as well as seamless integration with mobile robots.

3.5.3.1 ATLAS: Open-Source TDOA-Based Localization

In the context of the ATLAS Real-Time Localization System (RTLS), we propose building upon the TDOA topology in which the mobile nodes transmit a single message and the infrastructure receives (R-TDOA). Wireless clock synchronization is used to achieve a common time base among the clocks of the static infrastructure-based anchor nodes. Yet, random access is incapable of providing guaranteed update rates required by many robotic applications. Furthermore, with an increase of mobile nodes, the quality of the localization will degrade due to missed synchronization frames, as pointed out in [685]. This means that the real-time requirements for control-grade applications such as indoor UAV navigation cannot be met at scale by random access.

3.5.3.2 ATLAS FaST: Lightweight Scheduling for Control-Grade Applications

Based on previous work [684], we thus propose a novel lightweight scheduling protocol that seamlessly integrates with the wireless clock synchronization needed for proper operation, see [687]. As depicted in Figure 3.23, our approach aims to support scalable, low-power, reliable, and real-time capable wireless localization. In the following, we will document the approach and experimentally evaluate its capabilities in an industrial setting. The open-source implementation provided will enable the usage of scalable UWB localization in the robotics and automation community, see [678, 679].

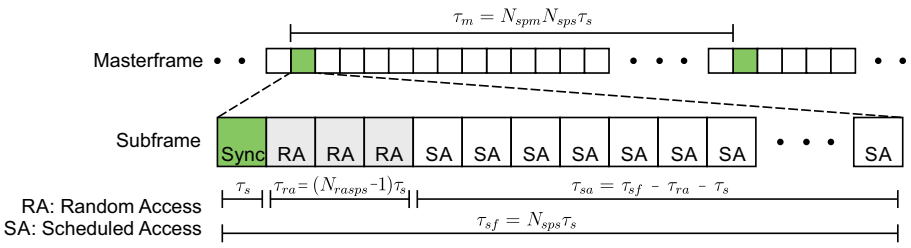


Fig. 3.25: Temporal structure of the ATLAS FaST localization specific Time Division Multiple Access (TDMA) scheme. The structure utilizing subframes enables multiple spatially distributed synchronization cells.

The publication [687] illustrates the continuity of the development of wireless localization building upon the widely used Robot Operating System (ROS). In contrast to previous work, which focuses mainly on plain localization aspects, a significant increase in scalability and real-time capabilities is provided, presenting a method that allows for scalable localization without degradation in the performance of the localization results.

The proposed system architecture is depicted in Figure 3.24. The ATLAS Concentrators are capable of connecting to multiple anchors and synchronize anchors through direct connection.

The chosen TDOA topology used in our system design enables our mobile nodes to be very energy efficient, as pointed out in Section 3.5.3.3. Only a single frame needs to be transmitted in order to obtain a full localization result. However, if a guaranteed update rate is desired, bi-directional communication, synchronization, and association with the system's scheduler are required. Through the modular system architecture, the individual components can be developed independently, which significantly reduces the time required to integrate application-specific features or competition-specific constraints. As mentioned before, the modularity of this concept is illustrated in Figure 3.24.

To lower the overall system complexity, the synchronization request is the same as any other positioning frame transmitted over the UWB channel. It consists of the tags Extended Unique Identifier (EUI) and a sequence number that is increased with

every new message. Depending of the application this message can be extended using Inertial Measurement Unit (IMU) data or battery status.

After accessing the channel, the tag immediately goes back to sleep and waits for the next sync frame. In the meantime, the anchor nodes receive and process the requests. Successful random access requests are propagated to the scheduling engine that has a database of pre-known period configurations and priorities from which it can prepare a response through the requesting tags EUI. This procedure simplifies the configuration overhead as the system is infrastructure-based. It also allows for seamless graceful degradation of the update rate if the overall system capacity limit is reached.

3.5.3.3 ATLAS Low-Power Timekeeping under Resource Constraints

For random access based schemes, proper absolute time-keeping is not required. For the proposed scheduled access though, knowing the correct absolute time is of the essence. However, maintaining the high-frequency clock of the transceiver modules is associated with significant loss of power and therefore, battery life. Due to this reason, a concept of time-keeping, synchronization, and constant calibration should be employed for proper low-power operation in the scheduled scheme. This basic concept can be utilized for real-world implementations in combination with ATLAS FaST to support future low-power applications.

The transceiver should run in a mode that does not enable the high-frequency clock sources and, thus, loses its absolute time knowledge. Therefore, the proposed scheme employs intelligent switching between different clock sources to maintain the transmission schedule in the mobile nodes.

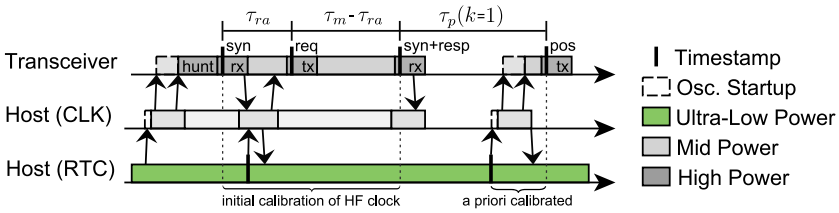


Fig. 3.26: An approach for low-power association with the system controller.

The initial association procedure at the mobile node is depicted in Figure 3.26. τ_{ra} is the duration between the sync frame and random access, $\tau_p(k)$ the duration between the response frame and the k^{th} positioning frame, τ_m the duration of the master frame, and k is the localization sample iterator. There are three main components of drawing power from the battery: the transceiver, the host controller, and a Real-Time Clock (RTC) within the host controller. Here, the internal RTC is used as the keeper of the overall system clock. The transceiver oscillator is calibrated using two consecutive

UWB synchronization frames. The resolution of the RTC is insufficient to calibrate upon initial association. Here, the re-association should be utilized as it provides a sufficient time-span enabling RTC drift calibration.

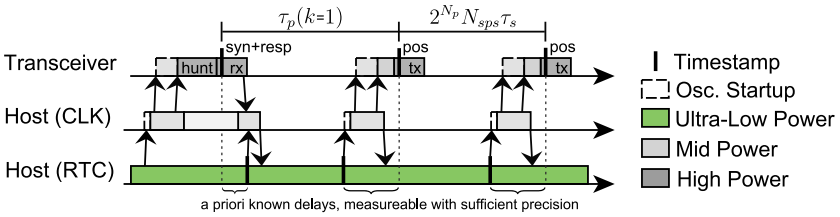


Fig. 3.27: Approach for low-power re-association with the system controller.

The typical re-association procedure, including the following transmissions, is depicted in Figure 3.27 where N_p is the positioning period exponent for a mobile node, N_{sps} the number of slots per subframe and τ_s the slot duration. The transceiver and the host-controller can be configured to a sleep state. An RTC timer can then be configured to wake up the host controller for the next positioning frame. Here, the delays for waking up the host controller and the transceiver need to be calibrated in the implementation phase. The processing times and oscillator start-up times need to be accounted for. Based on this, the RTC wake-up time can be configured. Although limited precision is given by the RTC, which mostly runs at 32.768 kHz, the available resolution of around 30.52 µs is sufficient to stay within the margins of the scheduling scheme. However, the potential error through clock-traversal needs to be accounted for in the slot-duration dimensioning for the scheduled scheme.

3.5.3.4 Wireless Signal Assessment to Improve Overall Accuracy

Due to the utilization of a TDOA-based localization scheme, another variable in the localization solution is introduced. Instead of rangings as with TWR-based localization, TDOA utilizes the time difference of the arriving signal. Hence, the localization is inherently more challenging. Therefore, countermeasures such as introduced in [680] are required to obtain precise localization results.

One of the main benefits of UWB is the availability of additional information of the received signal. Here, methods can be applied that leverage this information in order to weigh the individual measurements. Figure 3.28 illustrates this concept. Here, the ratio between the first path and the remaining energy of the channel impulse response is taken in order to weigh the measurements in the Extended Kalman Filter (EKF) used to obtain the localization results.

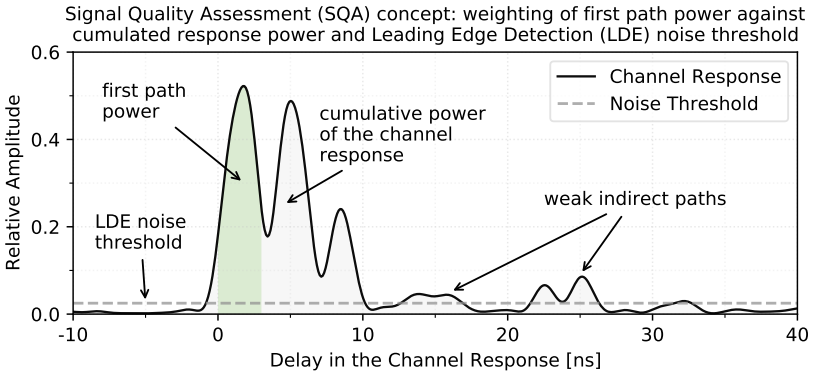


Fig. 3.28: An approach for signal quality analysis to improve accuracy. Note that the depicted channel response is interpolated from ten experimentally recorded sets in which the resolution is 1 ns per sample.

There are concepts that may be capable of extracting even more information such that even passive localization or simultaneous localization and mapping in certain scenarios are feasible. An initial evaluation of machine learning on these channel responses can be found in [686].

3.5.4 Results

In order to evaluate the performance characteristics of the presented approaches, selected evaluations based on the work in [683, 687] are discussed.

3.5.4.1 Scalability of the ATLAS FaST Approach

One of the main benefits of random access is the simplicity of implementation due to the lack of coordination. Tags can be implemented in a transmitter-only design, allowing for a long battery lifetime through intermediate sleep modes. The main down-side, however, is the lack of predictability. For many applications, especially in real-time control of autonomous systems, defined update rates are required. Here, a higher overall throughput is not directly useful, if long inter-arrival times are expected. Therefore, systematic channel access is desirable.

With increased loads, low energy, battery-powered applications suffer from the non-reception of localization frames. Therefore, the effective energy per position ratio increases, leading to decreased efficiency. The ATLAS FaST scheme is designed to overcome these issues. In the following, the achievable inter-arrival times and reliability are analyzed. $\hat{\tau}_{nf}$ considers a processing time of τ_{proc} and the partial preamble reception effect.

The resulting throughput is the superposition of these individual effects that depend highly on the implementation of the receiving side in practical systems. Due to the unique characteristics of the UWB PHY, a non-destructive R-TDOA scheme is feasible as the throughput does not degenerate with increased loads in the available ranges. However, due to the non-reception of frames, the real-time capabilities for the localization systems degrade as a defined update rate cannot be guaranteed. This is especially severe for applications, requiring tight real-time constraints.

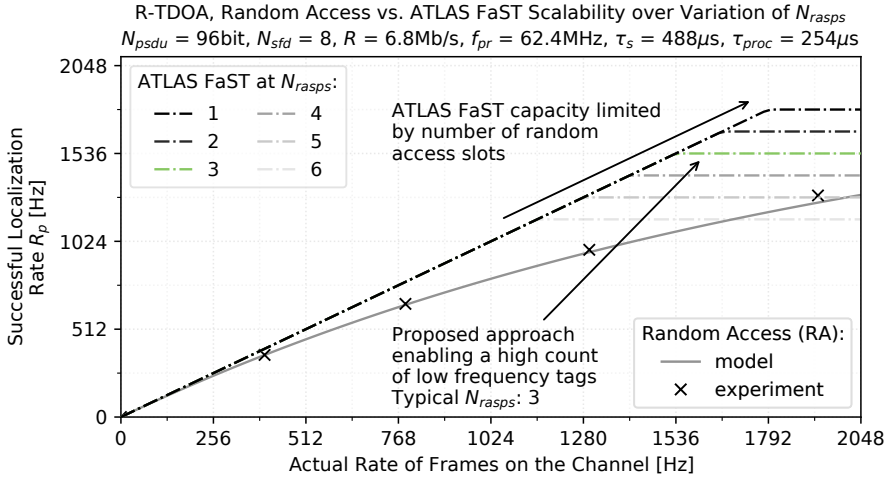


Fig. 3.29: Experimental vs. analytical throughput analysis for R-TDOA-based localization in which the infrastructure estimates the location.

ATLAS FaST overcomes this issue. Due to the slotted approach and the competition-free channel access in the scheduled slots, the successful positioning throughput scales linearly with the positioning frame load, as depicted in Figure 3.29. Here N_{psdu} is the number of transmitted payload bits per packet, N_{sfd} the UWB start of frame delimiter size in symbols, R the effective data rate, and f_{pr} the mean pulse repetition frequency on the physical layer. The slot duration τ_s is chosen such that there are 2048 available slots per second. Non-reception due to noise-induced frame error rates were neglected in this analysis as this will highly depend on the link budget the wireless localization system planner will provide for the given setup. However, the variation of the number of random access slots will define the upper bound of the system's capacity.

Alongside the throughput of the proposed FaST approach, Figure 3.29 depicts an analytical model for R-TDOA obtained in a scaled experiment. Here we can compare the multiuser scalability of the proposed approach with the previous, random access-based R-TDOA.

It is clear that the scheduled approach allows greater throughput than that produced by the same underlying positioning load using random access-based R-TDOA. Also, the analytical model for the R-TDOA throughput matches the experiment closely. The current implementation supports only static adjustment of the amount of random access slots N_{rasps}; for optimal performance a dynamic adjustment would be necessary. However, even when using a static set of six random access slots, FaST supports more than 1000 mobile units at 1 Hz considering a high re-association interval.

3.5.4.2 Energy Utilization of Localization Schemes

In order to provide a simplified comparison between the energy consumption at the mobile unit for different channel access schemes, the bar chart in Figure 3.30 summarizes the results of this work relative to the mostly used Symmetrical Double-Sided Two-Way Ranging (SDS-TWR)-based topology now considered state of the art.

Here, a minimal set of four anchors is the baseline for all topologies. Multiple different ranging or localization schemes are considered: R-TDOA with random access as introduced in the early ATLAS implementations [684]; T-TDOA as illustrated by [256, 378] with transmitting anchors and a receiving-only mobile unit, similar to GNSS; typical single-sided Two-Way Ranging (TWR), which requires a message exchange of two messages per ranging; TWR with Multiple Acknowledgments (TWR-MA), utilizing a repeated response to estimate clock offset; combined TWR, which orchestrates a TWR exchange by pre-defined individual response time offsets for a set of anchors; and, finally, symmetric double-sided TWR (SDS-TWR), which utilizes symmetric response times to cancel out clock drift during ranging. The last one requires three messages for basic SDS-TWR, but it can also be configured to report calculated ranges in SDS-TWR-R, requiring another message at the end of each ranging. For the SDS-TWR topology no reporting from the anchor side is considered. So the range information is available only at the infrastructure side as it is using R-TDOA or FaST. But SDS-TWR-R considers reporting and therefore consumes even more energy than plain SDS-TWR.

Keep in mind that the energy consumption of the TWR- and T-TDOA-based topologies increases linearly with the number of anchors in the setup, while the energy consumption of R-TDOA-based localization is independent of the anchor count. Therefore, it can be stated that even at the worst-case scenario the proposed topology outperforms traditional TWR-based schemes in terms of scalability and energy usage.

By calculating the power per transmitted and received UWB frame for the transceiver system and the fundamental requirements for the localization topologies, er can determine an idealized maximal battery life. An exemplary positioning rate of 1 Hz, a re-association period of 300 s, and a reliability look-ahead $N_r = 1$ were chosen.

The power consumption is calculated based on the datasheet of the DW1000 transceiver resulting in E_{tx}=40.7 mJ and E_{rx}=76.9 mJ. Note that the different PSDU (Physical layer Service Data Unit) sizes, that are required for the different schemes

are simplified to the lowest size of the R-TDOA scheme due to the implementation-dependency of this value. The discrepancy between TDOA- and TWR-based schemes would be even greater.

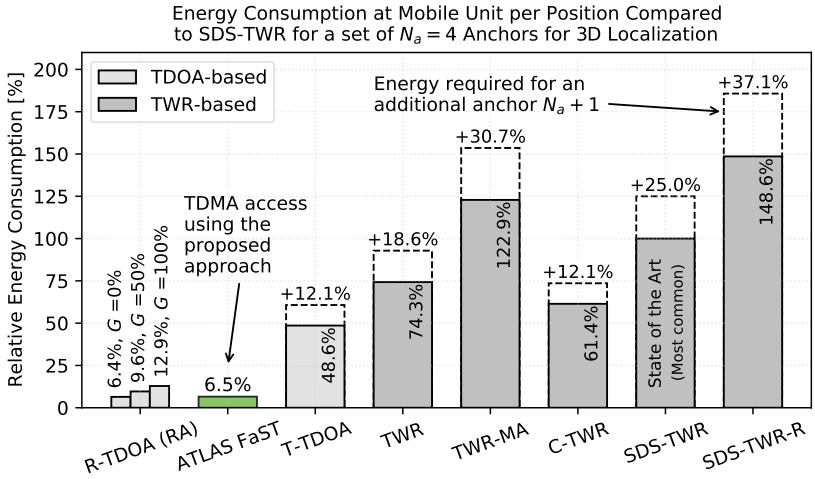


Fig. 3.30: Exemplary analytical evaluation of energy utilization per localization approach. Note that the main factor influencing the energy consumption is the amount of messages required per localization result.

The resulting battery lifetime is depicted in Figure 3.30. The battery lifetime of the T-TDOA- and TWR-based approaches is lower by orders of magnitude due to the dependency on the number of anchor nodes N_a inherent in those topologies. For the R-TDOA-based approaches, the FaST results are close to those of random access, which is the baseline for low power consumption at the mobile node because there is no additional overhead. Therefore, it can be stated that through the planned scheduling with infrequent synchronization, FaST is capable of tracking many low-power devices without interfering with the real-time requirements for critical applications.

3.5.4.3 Accuracy

For the evaluation of the accuracy, several tracks were followed. One of the main contributions were lab experiments evaluating the accuracy of the signal quality assessment under motion capture tracking as published in [680]. To achieve internationally comparable results, the participation in competitions is key, as it is the only way to benchmark localization results comparably. In the following, based on [683], the participation in two international competitions is highlighted briefly.

The EvAAL competition alongside the *Seventh International Conference on Indoor Positioning and Indoor Navigation (IPIN2016)* in Alcalá de Henares, Madrid, Spain is used to provide a comparable basis for localization systems in the context of robotic tracking. In the fourth track, *Indoor Mobile Robot Positioning*, six teams registered. While four teams registered *in-track*, two teams were *out-of-track* from within the organization team and were only evaluated for comparison. In [507, 549] the organization and results of this competition are covered in detail.

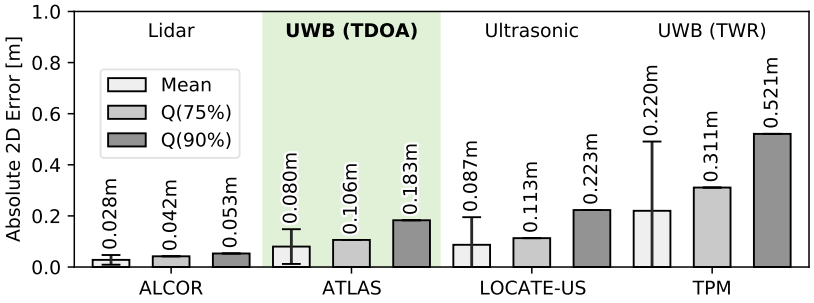


Fig. 3.31: Reconstructed competition results for the EvAAL 2016 as a bar chart. The official competition metrics were based on the third quartile Q(75 %).

The goal of the competition was to localize a mobile robot following a predetermined track that has to be obtained by the evaluated system. A set of four poles to mount the system under evaluation was provided by the organizers to cover the 12 m × 6 m evaluation area. The systems were evaluated sequentially so that one of the challenges of this competition was a maximum set-up time of 30 min that was extended to 45 min during the competition. The metric used for evaluation by the organizers is the third quartile of the Euclidean distance to the track, due to the missing temporal component of the evaluation setup.

A bar chart of the results is shown in Figure 3.31. In addition to the Q(75 %) quantile used for ranking, the mean absolute Euclidean error and the Q(90 %) error are depicted. It should be noted, that through the use of TDOA-based localization an additional unknown error is introduced, which generally lowers the accuracy of these systems. This has to be considered when comparing the ATLAS results and accuracy with the TWR-based system of the TPM team.

It can be seen that the ATLAS approach, although utilizing TDOA instead of TWR, can provide accurate robot tracking results and is, therefore, usable in the context of real-time robotic movement tracking.

To evaluate the localization approach in a more challenging environment, we participated in the fifth iteration of the *Microsoft Indoor Localization Competition* (MILC) co-located with the CPS-Week 2018 in Porto, Portugal. Previous competitions featured

broad participation from science and industry, see [422, 423]. In this competition, 34 teams submitted abstracts, 26 systems officially registered, and 25 systems showed up in Porto. However, only 22 systems could provide data and were evaluated.

The second category that used custom infrastructure such as UWB, was required to report 3D locations. Up to ten anchor nodes were allowed in the evaluation area. The teams had a time slot of 8 hours to set up and calibrate their systems.

The teams were evaluated using a mobile laser scanner-based ground-truth system. The organizers allocated a 15 min evaluation slot per team and fixed order. However, during the competition, the handling was more dynamic so that teams with a non-functioning system during their slot had the chance to debug their system and evaluate it later. Furthermore, since the competition area was the main staircase for the attendees of the four conferences held during the evaluation, the systems had to cope with the obstruction and interference of visitors and observers.

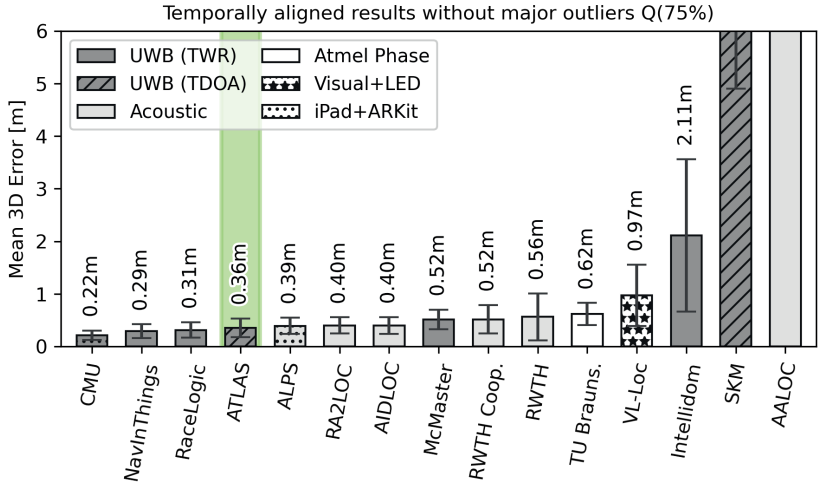


Fig. 3.32: Reconstructed bar chart of the mean 3D accuracies from the MILC 2018 data after temporal alignment and elimination of the 25 % largest errors for each team. Note the diversity of sensors for the individual localization methods.

The organizers chose the mean Euclidean error as the metric for evaluation. This error, however, is strongly influenced by the large outliers at the beginning of the evaluation run. Furthermore, a temporal offset of around 1 s between the ground-truth and the ATLAS-team result trajectory is observed. Due to this, additional errors were introduced into the evaluation.

Note that with improved temporal alignment, the mean error of the ATLAS team improves. Since it was desirable to compare the performance of the properly initialized

ATLAS results from the rest of the teams, the mean of the best 75 % of the temporally aligned results was evaluated. In order to provide a fair basis for comparison, the best 75 % of all teams were considered. Therefore, the analysis does not merely remove the outliers but improves the results of all teams as depicted in the lowest bar chart.

As depicted in Figure 3.32, when considering the aforementioned points, the actual performance of the ATLAS system is significantly better than indicated by the official results. Even though the ATLAS system uses a TDOA-based approach, which has significant downsides in accuracy, due to the proposed implementation, the results are comparable and often better than the other TWR-based approaches.

3.5.5 Conclusion

This section presented the basic capabilities, challenges, and novel solutions for high precision wireless localization. Here, the bounds in channel utilization, energy consumption, and achievable accuracy are limiting factors for the feasibility of a wide range of applications.

Therefore, novel approaches for maximizing information and localization throughput while minimizing channel utilization and power consumption and maintaining precise localization results were shown to overcome technology barriers and ultimately enable a connected cyber-physical world. The ATLAS FaST scheduling scheme and approaches for improving the achievable accuracy are highlighted.

It could be shown that based on these requirements, solution approaches are capable of improving both, channel utilization and energy efficiency. Further, approaches to increase the achievable accuracy in challenging environments can be successfully applied to improve TDOA localization accuracy.

In future work, challenges such as reducing the amount of required anchors, ad hoc configuration, and in-depth signal quality assessment with machine learning is envisioned. The potential for high precision localization enabled by the characteristics of novel communication solutions such as new UWB standards along with 5G and future 6G systems is huge.

3.5.6 Acknowledgments

In addition to the CRC 876, part of this work has been supported by the federal state of North Rhine-Westphalia and the “European Regional Development Fund” (EFRE) 2014-2020 in the course of the “CPS.HUB/NRW” project under grant number EFRE-0400008.

3.6 Indoor Photovoltaic Energy Harvesting

Mojtaba Masoudinejad

Abstract: Advancement in the field of electronics has enabled devices with ultra-low power (ULP) demands. However, Industry 4.0 devices made of such components will be empowered for long operational periods, specifically in remote or hardly accessible environments. Scavenging energy from the environment is a very common technique to tackle this energy supply issue. Different energy harvesting principles are available that exchange energy from distinct forms into electric power. However, we focus on Photovoltaic (PV) energy harvesting because it is the most mature technique.

In addition to the small size and weight limitation of Industry 4.0 and IoT devices, which require constraints on the size of a PV cell, they are applied mostly in indoor environments. Hence, the specific behavior of PV modules for indoor applications under artificial lighting is analyzed here. Using a systematic data acquisition procedure, typical PV models are adapted for the ULP harvesting environments. A normalization procedure is introduced during parameter tuning because common techniques are not applicable on PV modules when operating in low and ultra-low lighting conditions. Guidelines are provided to assure the numerical stability of parameter tuning of models.

The second layer of a two-fold model represents the relation of tuned curves with the environmental factors. Using a relative representation according to the highest light intensity, these models can be applied for different conditions. Performance of the overall model is evaluated on an extra dataset collected from a new environment showing model errors less than 6 % in the worst-case condition.

Parts of this section are taken from [434] with the consent of the author.

3.6.1 Introduction: Energy Harvesting

Energy is available in nature in a wide range of forms, from heat and mechanical energy to the energy stored in electromagnetic waves and light photons. Any method which enables scavenging these energies can be called *energy harvesting* and a transducer that converts them into the desired form is an energy-harvesting device, or *harvester* for short. Energy harvesting has a long history, since windmills, which convert wind energy into mechanical energy to mill grains, date back to the 9th century. Nonetheless, modern energy harvesting converts energy into electricity.

Wind, solar, Photovoltaic (PV), piezoelectric, thermal, radio frequency and tidal energy harvesting are only some examples of modern energy-harvesting techniques. However, techniques based on the conversion of light (specifically solar energy) into

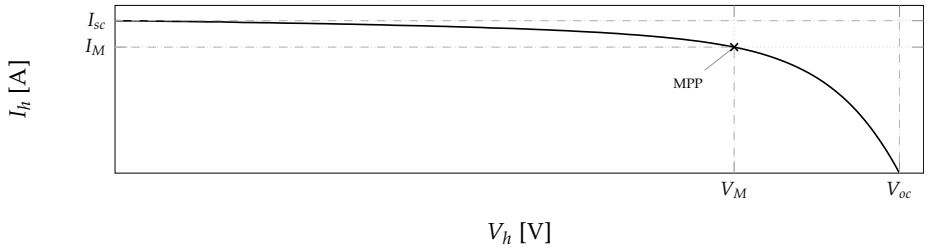


Fig. 3.33: I-V curve of a Sanyo/AM-1464 PV module measured under florescent light.

electricity are more mature than others due to long research and applications in diverse fields. While the term *solar* is commonly known, it includes different principles for converting the sunlight into the electricity. Nevertheless, PV is a specific form that uses semiconductor materials and technologies converting light into electric energy. In a simplified version, a PV transducer can be considered to use the inverse principle of a Light Emitting Diode (LED). In addition to the maturity of PV harvesting, their integration into the ULP Industry 4.0 hardware is the main reason we focus on them below.

3.6.1.1 PV Energy Harvesting

"A PV transducer is a semiconductor device generating electrical power when illuminated with photons [434]". These semiconductors have electrons in their valance energy band, which is weakly bounded. This bound can be broken by any photon that has higher energy than the band gap and causes movement of the electron to the conduction band. As long as enough photons are illuminated on the semiconductor surface, a photon's energy is converted into the flow of electrons to convert light into electric energy. Due to the nature of this conversion, PV can generate Direct Current (DC). Consequently, a PV harvester is an electric source, but not an ideal one. According to the operational condition of the PV module it can act as either a voltage or a current source. The common I-V behavior of a PV module is shown in Figure 3.33.

As can be seen in Figure 3.33, for a large portion of the voltages (mainly in the lower range) the PV module acts as a current source while for most current values (in a small voltage range) it acts as a voltage source. However, none of these sections are ideal because they are pure vertical or horizontal lines. The bending point where the behavior between source form changes is critical because the maximum power can be extracted from the module at this specific point. Hence, it is called Maximum Power Point (MPP) and techniques used to keep the operational point at this point are called Maximum Power Point Tracking (MPPT). When V_h and I_h describe the harvested voltage and current subsequently, MPP can be found mathematically from Equation 3.37.

$$\frac{\partial P_h}{\partial V_h} = \frac{\mathrm{d}I_h}{\mathrm{d}V_h} \cdot V_h + I_h = 0, \qquad \text{at: } V_h = V_M \tag{3.37}$$

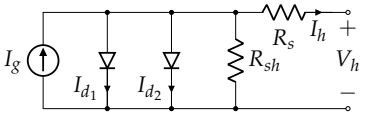


Fig. 3.34: Equivalent 2 diodes circuit model of a PV transducer including parasitic resistances.

In addition to the MPP, there are two other critical points for a PV module including the crossing from voltage and current axes: *open-circuit* and *short-circuit* values.

3.6.2 PV Transducer Model

While some applications require only knowledge of these three specific points, most utilizations of PV systems require a more descriptive explanation of the whole I-V curve. Based on the Shockley diode equation [62] as an outcome of work by Hall [254] and Shockley et al. [604], a multi-diode model of this behavior is explained as:

$$I_h = I_g - I_{d_1} - I_{d_2} - \frac{V_h + I_h \cdot R_s}{R_{sh}}, \tag{3.38}$$

when I_g is the photo-generated current, R_s and R_{sh} are subsequently series and parallel resistances. Diode's current I_d is defined as in Equation 3.39 when $\star$ is the diode number.

$$I_{d\star} = I_{s\star} \cdot \left[\exp\left(\frac{V_h + I_h \cdot R_s}{n_\star \cdot V_t} \right) - 1 \right] \tag{3.39}$$

In Equation 3.39, I_s and n are saturation current and ideality factor of each diode. In addition, V_t describes the *thermal voltage* from $V_t = B \cdot T/q$ when B is the Boltzmann's constant, T is the temperature in Kelvin, and q is the electric charge.

From the formulation in Equation 3.38, an Equivalent Circuit Model (ECM) can be made for the reproduction of the source behavior of a PV module. This ECM shown in Figure 3.34 is able to replicate characteristics of the PV cell, including its non-idealities and resistances. However, according to the desired accuracy of the I-V curve replication, it is possible to reduce the number of diodes into one. But some applications utilize higher number of diodes to increase the degree of freedom for a better replication of the curve.

Regardless of the number of diodes in a PV transducer's model, this model has to be fitted into the real curve of each module and light type. Consequently, the number of parameters to tune differs according to the number of diodes. Moreover, a system designer may even remove some parameters from the model in some applications. Therefore, a general vector $\boldsymbol{\psi}$ will be used to represent these unknowns.

3.6.2.1 Parameter Tuning

There is a large body of research on methods for tuning $\boldsymbol{\psi}$ for each PV module. However, they can be simply divided into *numerical* tuning and *Algebraic Equation Set* (AES)

tuning. In numerical methods, a set of points on the I-V curve is measured and directly used to solve an optimization problem to minimize the error. For an ideal case when the whole I-V curve is measured, this can be formulated as error minimization when $\hat{I_h}$ shows the current from the tuned model:

$$\min_{\boldsymbol{\psi}} \ell(\boldsymbol{\psi}) = \int_0^{V_{oc}} \parallel I_h(V_h) - \hat{I}_h(V_h, \boldsymbol{\psi}) \parallel_2 \, \mathrm{d}V_h \tag{3.40}$$

While this method can be explained in a simple way, its application has different challenges. At first, a large set of measurements from the PV system is necessary in a constant environmental condition. It can be especially challenging to keep the light intensity constant for the whole measurement duration. Other issues are related to the computational aspects of this method. On the one hand, a roughly computationally intensive optimization has to be solved to reach a set of reliable values for parameters in $\boldsymbol{\psi}$. On the other, these methods are sensitive to the initial values used for these parameters. Another issue that adds complexities to this utilization is finding the current value for each voltage in the model. As seen in the model's formulation, Equation 3.38 describes an implicit relation between voltage and current. Therefore, the calculation of current requires either an explicit relation or has to be done in a numerical (iterative) way. For the single-diode model, Femia et. al. [204] gives an explicit relation between parameters using the Lambert function as:

$$I_h = \frac{R_{sh} \cdot (I_g + I_s) - V_h}{R_{sh} + R_s} - \frac{n}{R_s} \cdot \mathcal{W}(\theta_i) \tag{3.41}$$

where Lambert W function is presented with $\mathcal{W}$, and θ_i is:

$$\theta_i = \frac{R_{sh} \cdot R_s}{n \cdot (R_s + R_{sh})} \cdot I_s \cdot \exp\left[\frac{R_{sh} \cdot R_s \cdot (I_g + I_s) + R_{sh} \cdot V_t}{n \cdot (R_{sh} + R_s)}\right] \tag{3.42}$$

Although this simplifies the single-diode finding of the current in the model, the two diodes model (which is more accurate) still requires a numerical solution for each voltage value. Therefore, there is a numerical iterative function solving inside the optimization in Equation 3.40, which is also solved in a numerical iterative sense itself.

The computational complexity of the numerical method is the main reason for most researchers to overcome this challenge by finding alternative routes for tuning parameters in $\boldsymbol{\psi}$. Using physical knowledge from the model is a common way to build an AES that can be solved with less computation. From the general I-V relation and basic knowledge about a PV system, several equations can be formulated:

$$\text{at SC: } V_h = 0 \Rightarrow I_{sc} = I_g - I_s \left[\exp\left(\frac{I_{sc} \cdot R_s}{n \cdot V_t}\right) - 1\right] - \frac{I_{sc} \cdot R_s}{R_{sh}} \tag{3.43a}$$

$$\text{at OC: } I_h = 0 \Rightarrow 0 = I_g - I_s \left[\exp\left(\frac{V_{oc}}{n \cdot V_t}\right) - 1\right] - \frac{V_{oc}}{R_{sh}} \tag{3.43b}$$

$$\text{at MPP: } I_M = I_g - I_s \left[\exp\left(\frac{V_M + I_M \cdot R_s}{n \cdot V_t}\right) - 1\right] - \frac{V_M + I_M \cdot R_s}{R_{sh}} \tag{3.43c}$$

It has to be noted that these equations are for a single-diode model but can be expanded for the double-diode model as well. While open circuit and short circuit can be easily measured, finding the MPP without any prior knowledge is not possible. However, it is known that the MPP is mostly proportionate to the open circuit voltage (V_{oc}) and short circuit current (I_{sc}). Hence, it is possible to simply measure the parameters at these specific proportional values and use them in Equation 3.43c, though it introduces a marginal error. The above equations are only a mathematical representation of the exact points from the curve, which is somehow similar to the numerical method. Hence, using the limited knowledge from this AES can provide a minimal fit to the model on these specific points. Furthermore, even a single-diode model has a $\boldsymbol{\psi}$ with 5 values. Consequently, this AES is under-determined and requires either more equations or a reduction of some parameters. Although few researchers have experimented on models with only 3 or 4 parameters, it is a well-established principle to use the derivation of the model on keypoints to expand the AES. These equations for a single-diode model are:

$$\left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{I_h=0} = -I_g \cdot \left[\frac{1}{n \cdot V_t} \cdot \left(1 + \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{I=0} \cdot R_s\right) \cdot \exp\left(\frac{V_{oc}}{n \cdot V_t}\right)\right] - \frac{1}{R_{sh}} \cdot \left(1 + \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{I_h=0} \cdot R_s\right) \tag{3.44}$$

$$\left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{V_h=0} = -I_g \cdot \left[\frac{1}{n \cdot V_t} \cdot \left(1 + \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{V_h=0} \cdot R_s\right) \cdot \exp\left(\frac{I_{sc} \cdot R_s}{n \cdot V_t}\right)\right] - \frac{1}{R_{sh}} \cdot \left(1 + \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{V_h=0} \cdot R_s\right) \tag{3.45}$$

$$\left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{\mathrm{MPP}} = -I_g \cdot \left[\frac{1}{n \cdot V_t} \cdot \left(1 + \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{\mathrm{MPP}} \cdot R_s\right) \cdot \exp\left(\frac{V_M + I_M \cdot R_s}{n \cdot V_t}\right)\right] - \frac{1}{R_{sh}} \cdot \left(1 + \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{\mathrm{MPP}} \cdot R_s\right) \tag{3.46}$$

Furthermore, the derivation of power according to the voltage is zero at MPP, which leads to Equation 3.47.

$$\left(\frac{\partial P_h}{\partial V_h}\right)_{\mathrm{MPP}} = V_h \cdot \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{\mathrm{MPP}} + I_h = 0 \Rightarrow \left(\frac{\mathrm{d}I_h}{\mathrm{d}V_h}\right)_{\mathrm{MPP}} = -\frac{I_M}{V_M} \tag{3.47}$$

Substituting this in Equation 3.46 adds Equation 3.48 as an additional equation to the AES.

$$-\frac{I_M}{V_M} = -I_g \cdot \left[\frac{1}{a \cdot V_t} \cdot \left(1 - \frac{I_M}{V_M} \cdot R_s\right) \cdot \exp\left(\frac{V_M + I_M \cdot R_s}{a \cdot V_t}\right)\right] - \frac{1}{R_{sh}} \cdot \left(1 - \frac{I_M}{V_M} \cdot R_s\right) \tag{3.48}$$

From all these equations, an AES with 7 equations can be made that are all explicit. Nevertheless, in addition to the complexity of measuring MPP, the challenging task of

finding the curve's slope at keypoints is mandatory. Some simplified solutions suggest calculating the slope from the slope of a line between keypoints, though it adds error to the tuning performance. Therefore, further measurements are necessary to improve the calculation of the slopes. Moreover, these equations fit the model's formula to the curve only for the measured keypoints and their derivation and cannot provide high accuracy on other points. However, it is possible to measure any further point from the curve and add it as a simple equation in the AES. As can be seen, both parameter tunings have their advantages and disadvantages.

3.6.2.2 Environmental Factors

An I-V curve represents a PV module in a specific environmental condition according to light intensity (E) and temperature (T). While the overall form of a curve remains, its details, including keypoints and slopes, shift with these parameters. Therefore, a further level of model is required to explain the effect of environmental factors. Some initial models explain the changes in keypoint values due to the deviation in environmental elements. Unfortunately, these models are not consistent in the literature and there are different formulations for a single parameter. However, there is a similarity in the methodology of these models due to their origin from PV operation in the solar light. In these models, environmental factors are described in a relative sense according to a reference condition, which is mostly the AM1.5 condition. It is described under a single sun at 1000 W/m^2 with a perpendicular line of sight to the PV cell at 25 °C.

Few relations are available in the literature, which explains some of the parameters in $\boldsymbol{\psi}$ according to the environmental factors. However, most of them use some kind of simplification assumption and mostly explain a specific application case study. Masoudinejad [434] reviews the state-of-the-art formulation for these parameters.

3.6.2.3 Indoor PV Energy Harvesting

Despite the maturity of PV energy harvesting and the availability of diverse techniques for analysis and modeling, most of the available methods are based on the applications under sunlight. By contrast, modern use cases, especially within the IoT and Industry 4.0 realm, are in indoor areas with artificial lighting. Consequently, a revision of the available methodologies and validation of their solutions is required. One of the major difference in these fields is the scale of light intensity. Figure 3.35 provides an overview. As can be seen, the light intensity in some industrial applications such as PhyNetLab [197] as an industrial warehouse is multiple orders less than solar-based applications. In addition to the light-intensity level, the indoor light spectrum has many more forms and can differ according to the light source, building materials, and surrounding environment. This difference in the spectrum between solar light and typical indoor light sources can be seen in Figure 3.36.

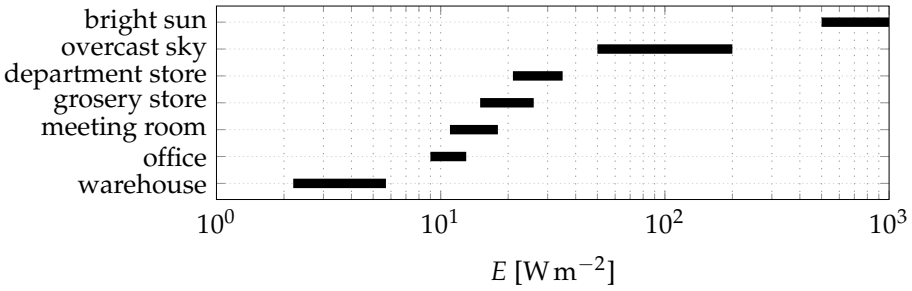


Fig. 3.35: Light intensity range in some common conditions. Reproduced from [146].

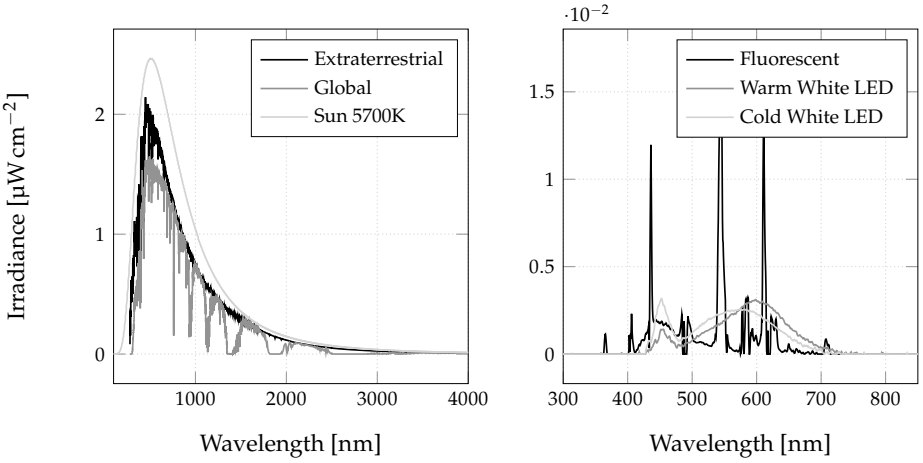


Fig. 3.36: Left: outdoor solar light spectrum [479] and sun's black body radiation at 5700 K [703]. Right: measured indoor light spectrum of three different artificial lighting.

Differences between light sources directly affect the I-V curve of a PV module. Figure 3.37 provides an example. Although both light sources are from the same manufacturer with the same specification and power, they produce dissimilar curves.

The only difference between these two sources is their color temperature, which can be seen in their spectrum in Figure 3.38.

As can be seen, although integrative light intensity of both sources is equal at 248 lx, their color temperature difference is a consequence of a discrepancy in the form of spectrum. Due to the non-uniform sensitivity of PV modules to each wavelength, the produced I-V curve will be different for each specific condition. Consequently, not only is an analysis of the PV behavior and modeling under artificial lighting necessary; careful consideration is required as well.

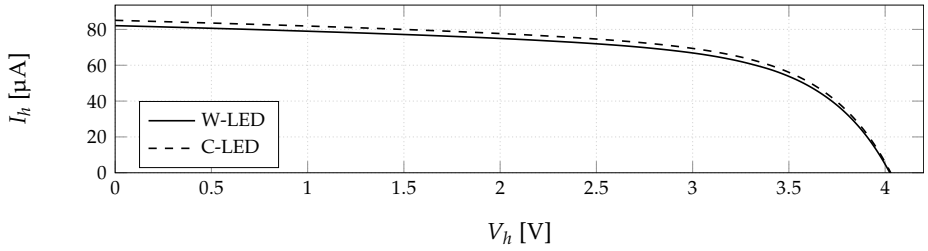


Fig. 3.37: I-V curve of a Solems PV module measured under cold and warm LED light. Both sources are from the same manufacturer measured at: E =248 lx and T =299 K.

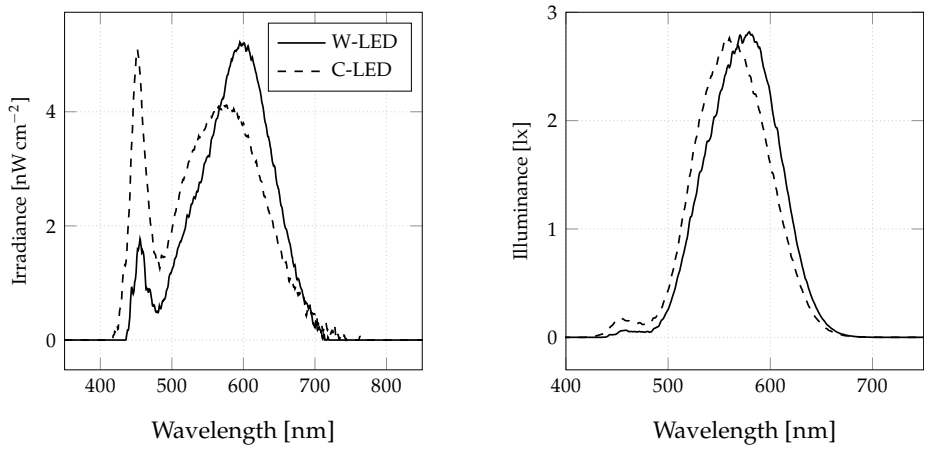


Fig. 3.38: Light spectrum measured under cold and warm white LED light both at 248 lx. In spite of equal integrative value, the spectrum is different. Left: irradiance, Right: illuminance.

3.6.3 Indoor PV Modeling

The diversity of the indoor lighting types and the lack of reliable data from PV behavior under indoor artificial lighting demand the collection of representative datasets for them. This data can be used later to analyze the I-V curve for such lighting, for parameters extraction through curve tuning and for formalizing the relation of them according to the environmental factors.

The required information for each measurement of the PV in an indoor area includes the I-V curve and light-intensity information in addition to the temperature. For measuring the I-V curve, a variable impedance has to be connected to the PV cell. Starting from a very large value this impedance will decrease till the open circuit voltage is reached. During this impedance sweep, voltage and current have to be measured simultaneously the whole time. This can be used to reproduce the curve. Such a procedure can be applied by using a Source Measurement Unit (SMU).

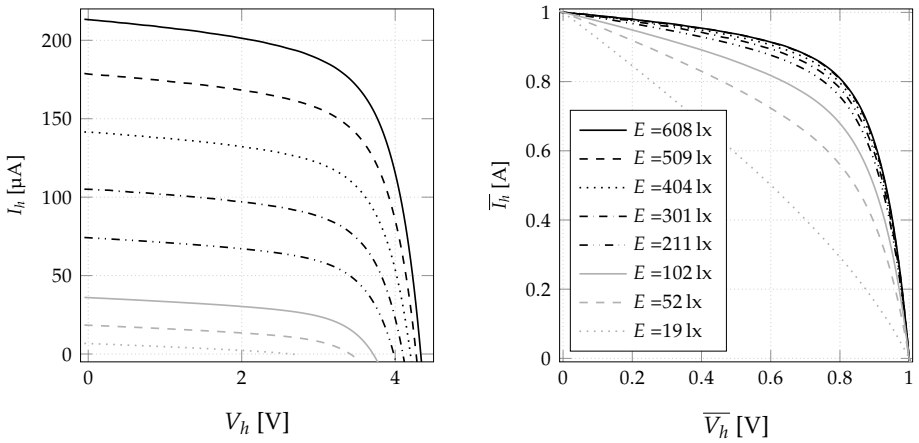


Fig. 3.39: Examples of measured I-V curves. Left: in normal space. Right: in PVNS.

There are multiple systems and devices for light intensity measurement. To have combined information, both integrative measurement and spectrometry are used here. Moreover, a closed environment with controlled light intensity is used to assure reliable and reproducible lighting. Using a system with detailed explanation from [432, 433, 435] different datasets are collected, which are publicly accessible from [433].

These datasets will be used below to develop the models and explore indoor PV behavior. However, there are two preliminary topics that need to be discussed before modeling.

3.6.3.1 PV Normalized Space

Looking at the available indoor datasets, as can be seen in the example in Figure 3.39, reveals that signal ranges are in different magnitudes. This diversity can cause calculation errors and add complexity to the numerical method calculations.

Therefore, *PV Normalized Space* (PVNS) is introduced to scale all curves into a similar range and avoid numerical problems. This conversion can be simply applied by converting the voltage and current of each curve according to its maximum referred to as V_{oc} and I_{sc}, respectively. It has to be noted that resistances have to be scaled as well, though because the parameter n does not have units it does not require any scaling. All in all, the relation of all scaled parameters (shown by $\overline{\bullet}$) is:

$$\overline{\boldsymbol{P}} = \boldsymbol{P} \times \left[\frac{1}{V_{oc}}, \frac{1}{V_{oc}}, \frac{I_{sc}}{V_{oc}}, \frac{I_{sc}}{V_{oc}}, \frac{1}{I_{sc}}, \frac{1}{I_{sc}}, \frac{1}{I_{sc}} \right], \tag{3.49}$$

when:

$$\boldsymbol{P} = \left[V_h, V_t, R_s, R_{sh}, I_h, I_g, I_s \right]^{\mathsf{T}}. \tag{3.50}$$

The effect of this scaling on the I-V curve can be seen in Figure 3.39 on the right. This scaling highlights the differences on the form of the I-V curve as a consequence of

Tab. 3.7: Bound suggestions for the PV model's parameters in the PVNS.

Parameter	Lower limit	Upper limit	Unit
I_g	0.9	1.1	[A]
R_s	ϵ	$(V_{oc} - V_M)/I_M$	[Ω]
R_{sh}	$V_M/(I_{sc} - I_M)$	∞	[Ω]
I_s	ϵ	1.1	[A]
n	1	10	[-]

different light ranges, which are not easily detectable on the original curve. Moreover, it brings all available data into a uniform scale which helps to reduce the sensitivity to the signal values in numerical methods.

3.6.3.2 Evaluation Criteria

Similar to the modeling procedures, an evaluation factor is required to quantify the performance of a model. When replication of the I-V curve is desired, relative performance factors are preferred due to the large signal range. Nonetheless, division to zero at the open-circuit point can hinder calculating a relative factor without removing this point. Unfortunately, open circuit is a keypoint and plays a critical role in any model. Consequently, the Mean Absolute Normalized Error (MANE) is defined here as in Equation 3.51. It normalizes the percentage absolute error according to the I_{sc} to find the mean value. This can be explained as the Mean Absolute Error for the I-V curve in the PVNS as well.

$$\text{MANE} = \frac{100}{I_{sc} \cdot V_{oc}} \cdot \int_0^{V_{oc}} |\delta(I_h)| \, \mathrm{d}V_h = 100 \cdot \int_0^1 |\delta(\overline{I_h})| \, \mathrm{d}\overline{V_h} \tag{3.51}$$

or for the case of discrete measured data with m points:

$$\text{MANE} = \frac{100}{I_{sc} \cdot m} \cdot \sum_{i=1}^{m} |\delta(I_h)| = \frac{100}{m} \cdot \sum_{i=1}^{m} |\delta(\overline{I_h})|. \tag{3.52}$$

3.6.3.3 IV Curve Parameter Tuning

In the first step of the modeling, $\boldsymbol{\psi}$ parameters have to be tuned for each I-V curve. For this purpose, SWL, SCL, and IWL datasets from [433] are used. For the numerical tuning method, as discussed in 3.6.2.1, the initial value used for each parameter plays a critical role. After a large set of experiments, a general guideline can be provided as in Table 3.7 for the bounds on these parameters in the PVNS. Using these bounds, parameters for all curves in datasets are tuned using least square method. This process is repeated for both single- and double-diode models using 200 equidistant points along the voltage axis, for each curve. Distribution of error for all cases is presented in Figure 3.40. As

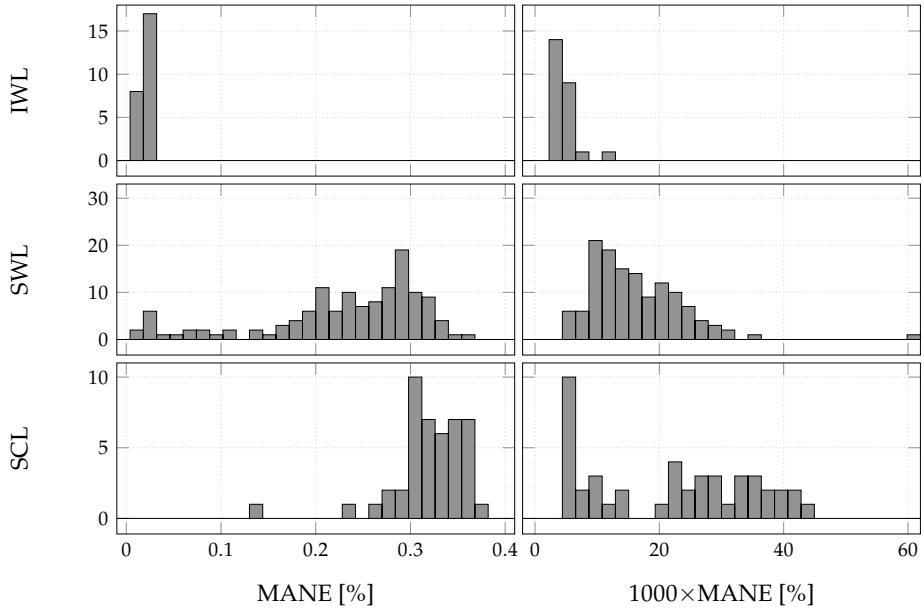


Fig. 3.40: Density of MANE distribution for all datasets using, the single-diode model (left) and double-diode model (right).

could be estimated, double-diode model has a much better performance compared with the single-diode counterpart. This can be simply argued because of extra tuning parameters.

The application of the AES-based method on these datasets is computationally simpler while it is very sensitive on the way that the slope of the curve is calculated. However, comparing the results with the numerical method, AES-based tuning has lower performance because of the extensive number of points in the numerical method. Hence, tuned parameters from the numerical method continue to be used. Their distribution according to the light intensity for the single- and double-diode models can be seen subsequently in Figure 3.41 and Figure 3.42.

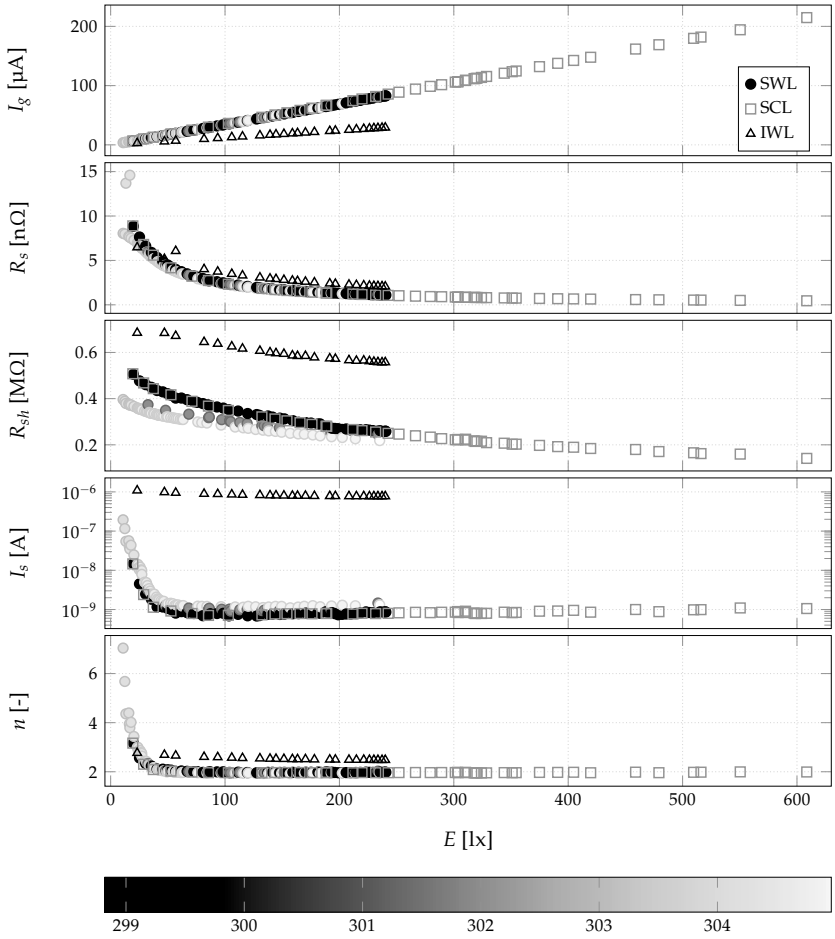


Fig. 3.41: Changes in single-diode model parameters according to the light intensity for all datasets. Color of SWL points shows temperature in K as in the color bar.

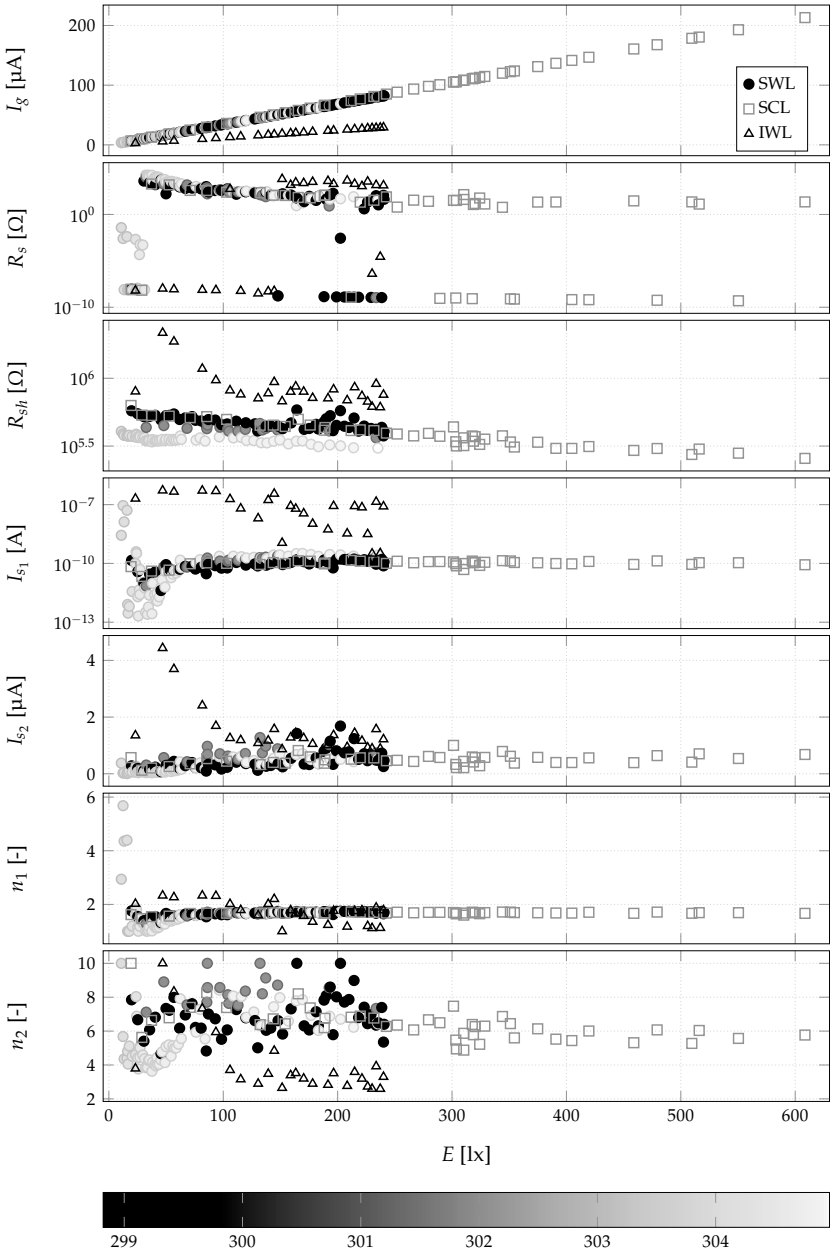


Fig. 3.42: Changes in double-diode model parameters according to the light intensity for all datasets. Color of SWL points shows temperature in K as in the color bar.

As can be seen from these figures, the single-diode model parameters have a clear relation to the light intensity and temperature. Although this holds for some parameters of the double-diode model, others such as resistors and saturation currents show abnormal behavior. This can hint at an over-dimensioning of the model with a second diode that introduces parameters that are not actually part of the physical system. Therefore, a hypothesis can be made that a secondary diode (or at least some of its parameters) is not part of the physical model of the PV system under artificial lighting. Hence, it improves the performance of the tuning but is no physically significant at the higher level. This behavior is somehow similar to the principle of over-fitting in machine learning, where the model focuses too much on the data, so that the overall form gets lost. Nevertheless, this hypothesis cannot be proven at this stage (using available data) without any further physical insight. Consequently, the single-diode model will continue to be used here for the remaining part of the modeling of the indoor PV system.

3.6.3.4 Modeling Effect of Environmental Factors

After tuning all parameters for the single-diode model, the next step is to formulate the relation of each parameter with the environmental factors. However, since this is a purely data-based modeling approach, the resulting models are empirical and will not assure any physical insight. Yet, by building these models for all datasets that include different PV technologies and light sources, the generalization of the model can be kept to some extent. After testing diverse function types on the available data, these relations are formulated as:

$$\widetilde{I_g} = \frac{I_g}{I_g^\star} = \alpha_{g1} \cdot \widetilde{E} + \alpha_{g2} \cdot \Delta T \tag{3.53a}$$

$$\widetilde{R_s} = \frac{R_s}{R_s^\star} = \frac{\alpha_{s1} + \alpha_{s4} \cdot \Delta T}{\alpha_{s2} + \alpha_{s3} \cdot \widetilde{E}} \tag{3.53b}$$

$$\widetilde{R_{sh}} = \frac{R_{sh}}{R_{sh}^\star} = \frac{\alpha_{p1} + \alpha_{p6} \cdot \Delta T}{\alpha_{p2} + \widetilde{E}} + \ln\left(\alpha_{p3} \cdot \widetilde{E} + \alpha_{p5}\right) \tag{3.53c}$$

$$\widetilde{I_s} = \frac{I_s}{I_s^\star} = \alpha_{i1} + \alpha_{i2} \cdot \widetilde{E} + \frac{1}{\alpha_{i3} \cdot \widetilde{E}^{\left(\alpha_{i4} + \alpha_{i5} \cdot \widetilde{T}\right)}} + \alpha_{i6} \cdot \Delta T \tag{3.53d}$$

$$\widetilde{n} = \frac{n}{n^\star} = \alpha_{n1} + \alpha_{n2} \cdot \widetilde{E} + \frac{1}{\alpha_{n3} \cdot \widetilde{E}^{\left(\alpha_{n4} + \alpha_{n5} \cdot \widetilde{T}\right)}} + \alpha_{n6} \cdot \Delta T \tag{3.53e}$$

In these equations, parameter α shows a tuning factor for each dataset, while $\widetilde{\bullet}$ is a representation of a relative parameter. Each relative parameter under a reference condition is similar to the AM1.5 for the solar case. However, contrary to the solar case with a unique reference point, there is no constant condition that can be defined for all indoor environments. Hence, the point with maximum light intensity in each dataset is used as a reference point and all its parameters are used as a base condition. Despite

parameters whose relative value is found from a ratio, the difference of temperature is more suitable for these formulations and is used in Equation 3.53.

Using these formulations, $\boldsymbol{\alpha}$ is tuned for each dataset. As an example, the output parameter for each light intensity in the SWL dataset is presented in Figure 3.43. It is clear that these provided empirical models are able to replicate the behavior of each parameter with a good accuracy.

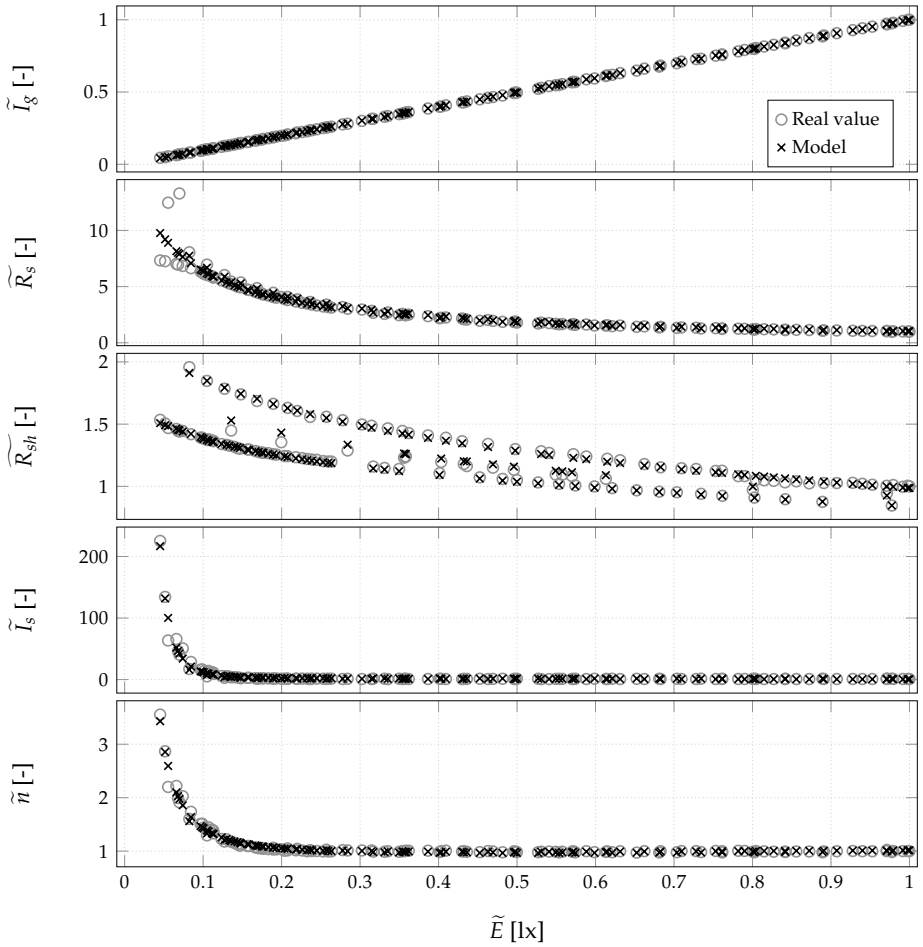


Fig. 3.43: Performance of models for parameters at different environmental conditions on the SWL dataset including temperature effect.

It has to be noted that parameters of the I-V curve with maximum light intensity are used here as the reference point. Also, it is advisable to remove the temperature factor

for a dataset when the deviation of temperature is minimal and can be ignored. In this way, the model can be simplified without reducing its performance.

3.6.4 Indoor PV Model Evaluation

So far, three available datasets have been used for the parameter tuning and environmental factor modeling. Since the overall behavior of the models has been checked on the same dataset, it can not guarantee a reliable performance for other conditions. Therefore, an extra evaluation on a new dataset is beneficiary. Consequently, a new set of data has been collected in the PhyNetLab, including 120 samples collected in different positions, heights, and temperatures from different days. This set includes light intensities between 244 lx to 494 lx and temperatures in the range of 298 K to 302 K.

To avoid error due to the training during the tuning of $\boldsymbol{\alpha}$ factors, 30 % of samples are selected randomly in a uniform distribution of the light intensity. Parameters for this smaller set is extracted and used in a least square method to tune $\boldsymbol{\alpha}$ in Equation 3.53. Similar to the model itself, the highest light intensity in this subset is used as the reference condition. The resulting formulations are used to find $\boldsymbol{\psi}$ parameters for the remaining 70 % of the data. Using these parameters, the resulting I-V curve is compared with the real measured value to find the MANE presented in Figure 3.44.

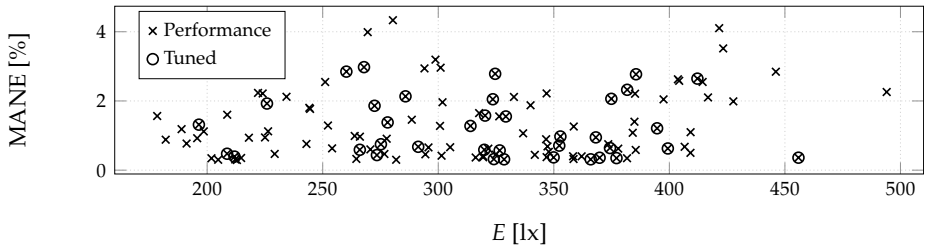


Fig. 3.44: MANE of model on the evaluation dataset. Only 30 % of data (randomly selected, shown with circles) was used for parameter tuning.

As can be seen, all errors are in a very small range, which shows reliable performance from the empirical abstract level model on a new indoor environment. When repeating the random selection subset, the worst case MANE is always less than 6 %. Hence, the overall modeling principle can be accepted and applied to other environments.

3.6.5 Conclusion

This contribution has discussed the basic principles of the PV energy harvesting and their modeling. After a short review of available PV models, we introduced tuning and adapting a model for a PV transducer. We then discussed the differences between PV behavior under solar light and artificial sources (used for indoor environments). Next we determined the need for model development and evaluation specifically for indoor PV harvesting.

It was shown that each light source and indoor environment has its particular specifications due to differences in the light spectrum seen by a PV module. Therefore, a new setup was presented to assure reliable, high accuracy, and reproducible indoor PV behavior data. Using this data, a PV normalized space was introduced to enable a unified modeling strategy, regardless of large differences in the signals within the indoor environment. Furthermore, the mean absolute normalized error was introduced as a non-compromised evaluation factor for model performance of each I-V curve.

Using the collected data, guidelines were provided for the boundary of tuning parameters in I-V curve fitting. While parameters were tuned for single- and double-diode models of a PV system, it has been shown that some parameters in the double diodes model lack specific relation to the environmental factor. This led to a hypothesis that the double-diode PV model includes more parameters than its real physical factors. Therefore, these parameter(s) improve the fitting but with the price of loosing the physical meaning. Hence, single-diode model data was used to develop empirical models for each parameter according to the environmental factors. These models use relative values according to a reference point which is simply selected as the data point with the highest light intensity.

Finally, these models were tested using all available datasets, and a new set of data was collected in a real-case scenario including 120 samples. The environmental condition of this new scenario was tuned into the model using only 30 % of data points selected randomly. The application of this tuned model showed promising performance with MANE of less than 6 % in the worst-case scenario. Therefore, we showed that adaptability of the developed models on new real world environments and their good performance on predicting the I-V curve of a PV module in new environments.

3.7 Micro-UAV Swarm Testbed for Indoor Applications

Nils Gramse
Moritz Roidl
Shrutarv Awasthi
Christopher Reining

Abstract: This contribution describes a testbed for the development of resource-constrained micro-UAV (Unmanned Aerial Vehicle) swarms . The testbed architecture combines external observation, visualization of logistic scenarios, and simulation systems with a drone control unit. Swarm algorithms are implemented on the drones to control their movements and enable their cooperation. A drone learns to perform a warehousing task using reinforcement learning. In combination with swarm algorithms, this behavior is extendable to a drone swarm. This work describes how drones can be deployed to solve tasks in industrial settings. In addition, an automatic charging station extends the runtime of the swarm.

3.7.1 Introduction

Drones are more formally known as unmanned aerial vehicles (UAVs) or unmanned aircraft systems. They are battery-powered devices, that can be as large as an aircraft or as small as the palm of a hand. A drone is an intelligent flying system with sensors and actuators that can be remotely controlled or fly autonomously. Due to its high adaptability, the use of drones is increasing across many civil application domains. Over the last decade, extensive research has been performed on consumer and commercial drones. Due to the advances in microelectronics, sensors are steadily getting lighter, smaller, more economical, smarter, and more accurate. As a result, drones are getting smaller, more energy-efficient, and easier to operate [521]. Moreover, there is a growing need to build small and resource-constrained drones that fit perfectly into the Industry 4.0 setting, where all intelligent devices are networked and can exchange their data with each other [52].

3.7.2 Drones in Logistics

Drones have shown high potential in the logistics industry. Some have projected that this market will grow by $29 billion by 2027 with an annual growth rate of almost 20 % [721]. A 2018 study found that electric drone delivery was more efficient than trucks, vans, passenger cars, and gasoline drones [650]. Electric drones are environmentally

friendlier than other aerial vehicles when comparing CO_2 emissions. Drones are a potential substitute for traditional delivery methods, such as door-to-door or last-mile delivery using trucks. The difference is that they do not consume fuel and do not need infrastructure such as streets. However, drones still pose several safety hazards, limiting their use in industries and outdoors. Some common safety hazards include drones colliding with people, structures on the ground, and other drones [720]. Thus, more research is required to make drones safe to use in urban and industrial settings. Deploying drones for transporting goods to end users may transform the existing transportation methods in large cities and rural areas. They might not completely replace the traditional methods but will significantly impact this domain [521].

Warehouse management is an essential operation for most business activities nowadays. Manual inventory has been the only option for a long time, but it poses several challenges in terms of costs, inaccuracies, and safety [151]. Furthermore, in the EU, warehousing and storage represent up to 15 % of the current costs in logistics [193]. Thus, there is a growing need to automate warehouse operations while ensuring the warehouse flexibility and adaptability. Such automation is something automated storage solutions made of mechanical conveyors such as high-bay warehouses and automated small parts storage cannot achieve. However, drones could potentially cater to this ever-increasing demand. There has been extensive research on drone use in transporting goods outdoors, and the researches are available commercially [418, 642]. However, large-sized drones cannot reach small, constricted spaces such as warehouses. The key reasons are that GPS-based navigation systems are unavailable, and the tolerances regarding collision avoidance and the time available for decision-making are drastically lower. By contrast, small-sized drones can operate in the interiors of warehouses and the narrow and high rows of shelves. Thus, one can integrate small-sized drones into the supply chain to automate various intra-logistics operations. [525].

Research on deploying autonomous drones and robots for human-machine interaction in warehousing is still in its early stages. However, industries have pioneered drones as extensions to their IoT environments or complement other data-gathering processes. Most IoT devices have a limited battery life, are stationary, and thus can collect the data of a specific location for a limited time. However, resource-constrained small drones can gather data from dangerous areas and other fixed IoT devices. Subsequently, they can transmit all the collected data to a central station. Drones can also aid in recharging IoT devices wirelessly or remotely. Thus, drones can be essential in connecting IoT devices to the whole IoT ecosystem [21, 26, 417]. This work introduces a testbed to replicate and experiment with any industrial scenario involving small-sized resource-constrained drones working alongside humans. In addition, an automatic charging station has been proposed, to ensure the continuous operation of drones.

3.7.3 Application

A possible logistics application for resource-constrained drone swarms, especially those that can recharge independently, is their use as a mobile surveillance system. For instance, this drone swarm can be used in warehouses to protect the stored goods from theft. The drones can then perch like birds on the existing trusses, high racks, and steel beams and observe their surroundings. From their high positions, they can gather a larger amount of data instead of low-power stationary IoT devices equipped with a battery. A charging station enables the continuous operation of the drone swarm. If a drone is low on battery, it automatically flies into the charging station, and another fully charged drone takes over its tasks.

3.7.4 Swarm Algorithms

Swarm algorithms create a group dynamic similar to the swarm behavior of animals, such as a flock of birds or a school of fish. Craig Reynolds developed a simulated swarm behavior in 1987 with the help of these observations [526]. In logistics systems, a swarm algorithm is similar to traffic rules, which enable a smooth traffic flow by observing the local environment with as little communication as possible between the vehicles. One advantage of using swarm algorithms in logistics is that they do not require fixed routes but can use all available spaces. This allows them to easily change the layout and react to unpredictable events or disruptions. Furthermore, due to the modular design of swarm algorithms, new behaviors can be added without altering the existing ones. The paradigm of swarm behavior aims to enable several independent robots to collaborate towards achieving a collective goal, acting as a swarm.

Swarm behavior, established by Reynolds, is implemented by three basic rules: the cohesion rule, the alignment rule, and the separation rule [153]. Figure 3.45 shows the rules schematically. The active agent in each case is the highlighted black triangle with its current velocity vector. The remaining triangles represent other agents. The highlighted velocity vector indicates the resulting vector adapted to satisfy the requirements of that rule. The white circles visualize the effective range of local perception of the active agent. The black circle represents the center of all agents located in the local perception.

The cohesion rule aims to move toward the swarm's center. Thus, an agent tries to point its velocity vector to the center of its local perception. The alignment rule ensures that all agents within the range of local perception aim for the same direction of motion. Finally, the separation rule states that a minimum distance to all neighbors [153] must be maintained.

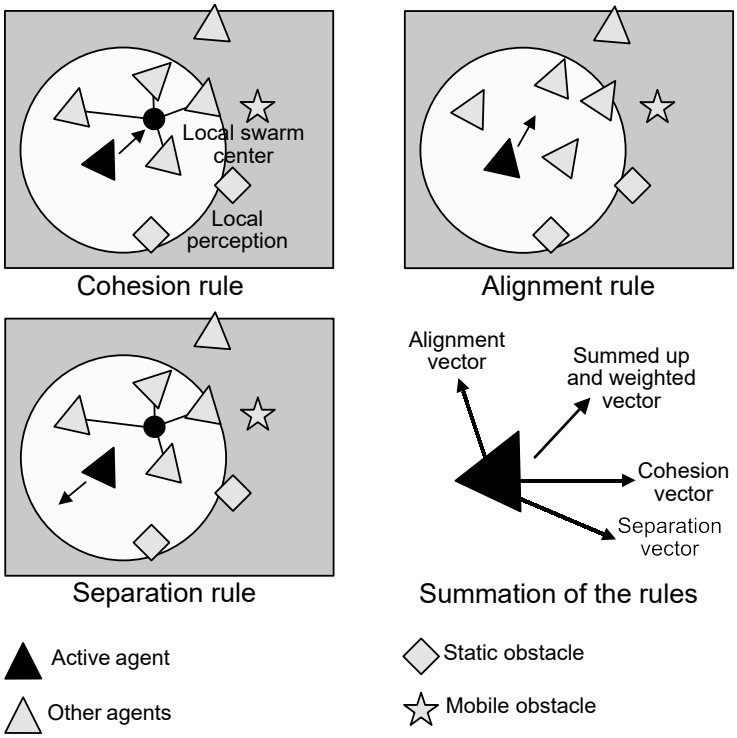


Fig. 3.45: Schematic representation of the three basic rules for simulated swarm behavior [153]. Mobile obstacles are people moving with headgear, and static obstacles could be the charging station, walls, or shelves.

Weighting the individual rules influences the appearance of the swarm behavior. For example, if collisions must not occur under any circumstances, the separation rule receives a high weighting. After assigning appropriate weights, the calculated velocity vectors are summed up to a control vector, which determines the final movement of an agent, as can be seen in Figure 3.45. The above three basic rules can also be supplemented by other rules as seen in [483]. One of the advantages of swarm control is that there is no need for the centralized control of each drone. Depending on the implementation, each drone in the swarm can perceive its local environment by itself and avoid collisions. Thus, it is possible to manoeuver an arbitrary number of drones simultaneously without calculating a separate trajectory for each drone centrally. This decentralized control makes the swarm implementation scalable.

3.7.5 System Architecture of Drones

The drones used in this work are a custom design using the Crazyflie 2.0 as a base platform [509]. Figure 3.46 shows the UAV used in this work. A single drone has a size of 103x103x29 mm and a total weight of 37.85 g. It can carry a payload of 64.85 g (compared with 15 g for an original Crazyflie drone). Thus, drones can use larger batteries which increases the flight time. Furthermore, additional sensors such as cameras and Lidar can also be mounted on the drone.

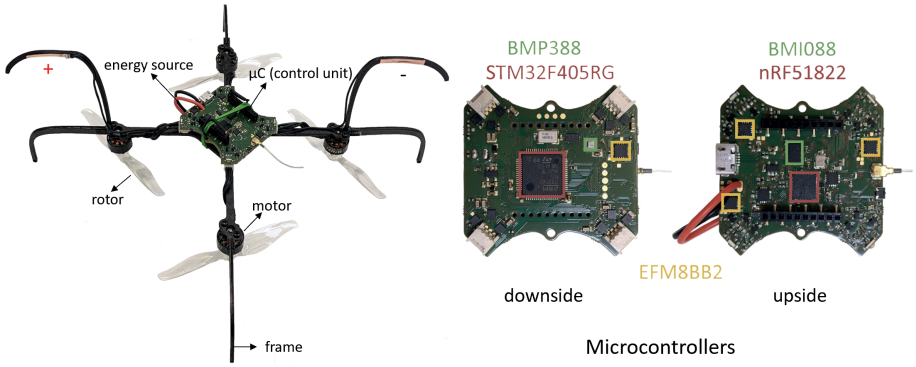


Fig. 3.46: Crazyflie 2.0 drones used in this work.

Four brushless DC motors on which four rotors with a diameter of 60 mm are mounted in reverse, comprise the propulsion system. Thus, the flow characteristics of the propellers are not influenced negatively by the frame. In addition, the frame structure itself is lightweight and built using fibre composite material.

The computing hardware consists of two microcontrollers. An STM32F405RG from STMicroelectronics [648] is used as the main computing unit. It performs all computationally intensive calculations and control tasks of the drone. For this purpose, a 32-bit ARM Cortex M4 with a clock frequency of 168 MHz and a floating-point arithmetic unit are used on the microcontroller. The SRAM has a capacity of 192 kB, and the flash memory holds 1 MB of program code. An EEPROM of 8 kB is connected to the microcontroller via the I^2C bus to store static information.

A second microcontroller (nRF51) from Nordic Semiconductor is used for wireless communication and as a power manager. This microcontroller uses a 32-bit ARM Cortex M0, which clocks at 32 MHz and has a 32 kB SRAM and a 128 kB flash memory. It has a low idle power consumption of 3 µW [591]. The two microcontrollers communicate via a UART interface.

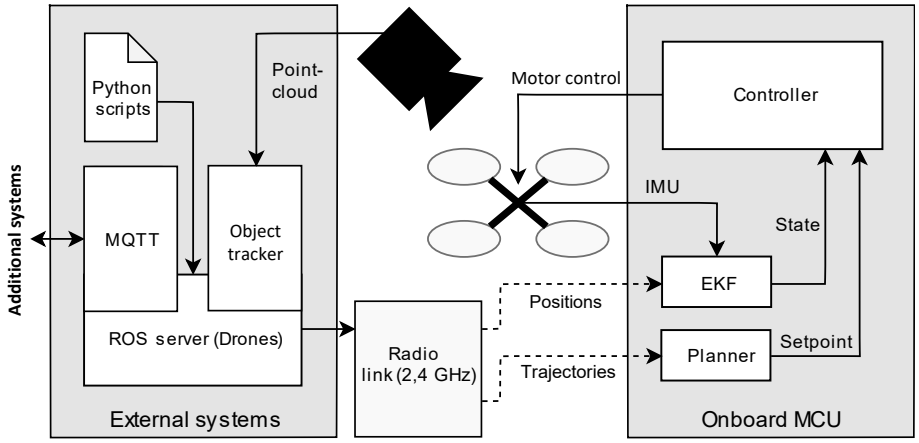


Fig. 3.47: System architecture for a drone swarm as a platform for data acquisition. The main components with their interfaces and the interaction between all components are shown [509].

Functionalities of the entire system architecture of the drone swarm are realized on various external systems, as seen in the Figure 3.47. Due to the loose interconnection of the multiple systems in the architecture, they can be easily replaced.

The Robot Operating System (ROS) server is well suited for inter-process communication [457]. Therefore, it is a part of the high-level application layer for controlling Crazyflie 2.0. Data from the MoCap system determines the absolute position and the attitude of the drones. A unique marker configuration consisting of four markers is pasted on every drone to identify each drone uniquely. The captured point cloud from the MoCAP system generates set points for the drones and compensates for the drift of the inertial measurement unit onboard the drone. All parameters converge in a ROS server and are forwarded to the drone.

The main computation task performed onboard the drones is the calculation of the flight parameters. A trajectory is computed from the set point and the state estimates in a specified time. The set-point values are obtained through an Extended Kalman Filter (EKF) to achieve a higher overall control accuracy. Trajectories can also be calculated externally and transmitted directly to the drone.

Communication between the external systems and the drone happens via the 2.4 GHz ISM band. The CRTP (Crazyflie Real-Time Protocol) is designed to send data packets without much overhead, thus minimizing the latency. The transmission speed is configured to 250 kbit/s to prevent interference caused by the metal walls of our research center and enable the transmitted signals to be decoded reliably by the drones. The MQTT server transmits drone information from the ROS server to other subsystems.

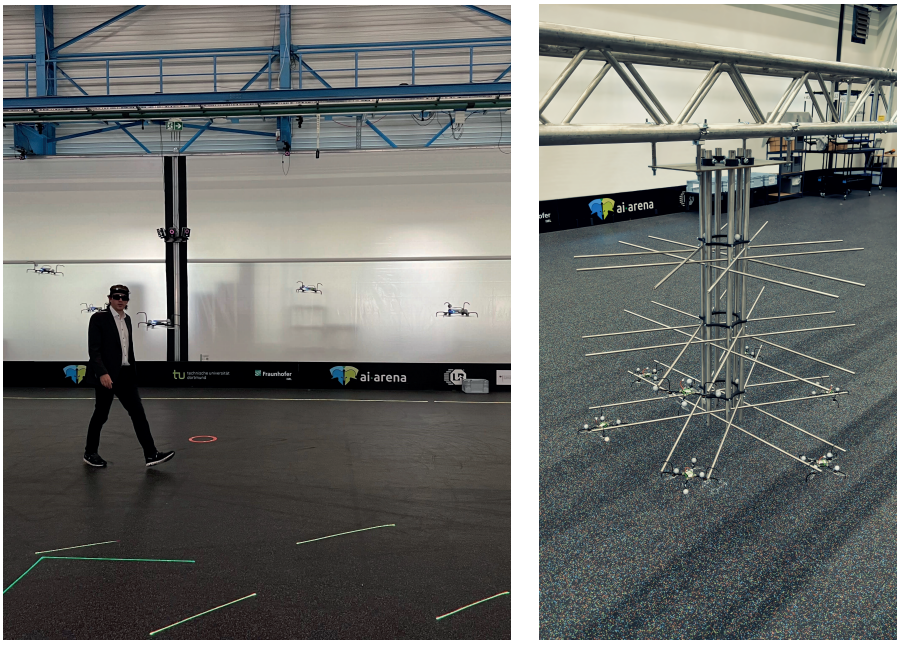
3.7.5.1 Recharge Station

Another adaptation is having the drones charged in a drone recharge station. Therefore, two holders with which the drones can hang in the station for charging, are attached to the drone frame. The charging current for the drone then flows via these holders. Thus, a battery exchange is no longer necessary, enabling continuous operation.

The drone recharge station represents another external system. It is constructed out of stainless steel rods and currently allows charging up to 32 drones distributed over three floors. A 200 W 5 V/20 A power supply feeds power to the recharge station. The recharge station can be hung from the roof or fixed in another safe place. When the drones land on the station, the charging of the drones starts automatically. Therefore, the 24/7 operation of the drone swarm is achievable. If a drone is low on battery, it automatically flies into the charging station, and another fully charged drone takes over its tasks.

3.7.6 Testbed

The testbed was developed at the Chair of Material Handling and Warehousing research center at TU Dortmund University. It is housed in a lightweight hall that is structurally identical to conventional industrial buildings in the logistics sector and follows the concept of a highly flexible development laboratory [478, 520]. Developing the testbed aims to create an environment for simulating real warehouse scenarios. The testbed is equipped with an infrastructure designed to prototype Cyber-Physical Systems (CPS) accurately. While the test area remains free of permanently installed infrastructure, the hall contains several permanently installed observation systems on the ceiling, walls, and floor.



(a) (b)

Fig. 3.48: General view of the testbed created in the research hall. (a) Person (wearing a headgear) standing between the drone swarm during a test scenario (b) Charging station hanging from the roof.

3.7.6.1 Architecture

Figure 3.49 shows the architecture used to create the testbed. The architecture is similar to the one used in [465]. The MoCap system consists of 46 infrared cameras manufactured by Vicon. It can track a substantial amount of appropriately marked objects with an accuracy of 0.3 mm and operates at a data transmission rate of up to 200 Hz with latencies of 4 ms to 15 ms. The observed experimental space is 22 m long, 15 m wide and up to 3.5 m to 4 m high. The localization data is accessible to multiple clients simultaneously over the network and provides the absolute position and attitude of the marked objects in a three-dimensional space. Figure 3.48 shows a general view of the testbed.

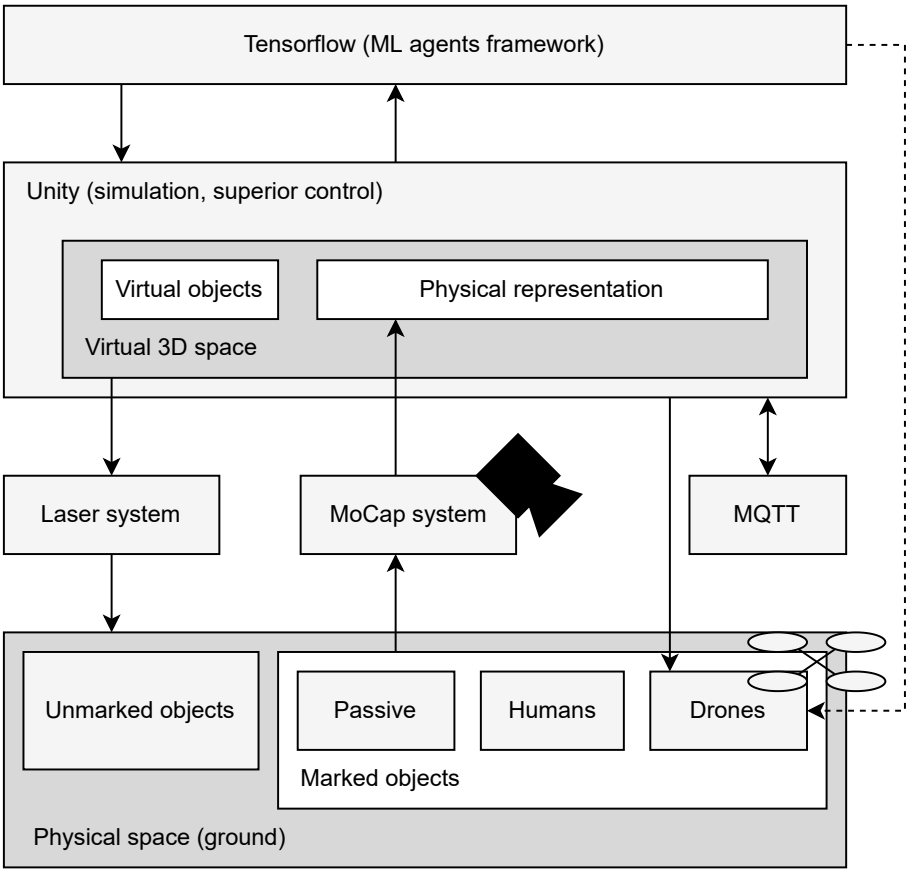


Fig. 3.49: Architecture used for creating the testbed [465].

The testbed uses MQTT (Message Queuing Telemetry Transport) [292] to distribute data between the subsystems in Figure 3.47. The simulation subsystem, designed as a development platform, consists of a programmable 3D modelling environment. In this work, Unity is used for 3D modelling. Unity generates virtual objects and scenes. All objects in a scene are mapped into an inheritance tree, which can be manipulated or extended using the C# programming language [307]. In addition to the common objects of the 3D modelling environment, such as cameras or light sources, the C# script creates custom objects.

The simulation subsystem comprises a laser projection system consisting of eight Kvant Clubmax FB4 laser projectors [373]. The laser system generates both static and dynamic projections of virtual objects from the simulation. Therefore, it is possible to visualize complex algorithms [436]. Visualization can be done for demonstration purposes and a better understanding of the complex behavior of an algorithm.

All marked objects in the physical space are mirrored into the Unity 3D simulation environment via the MoCap system. In addition to the representation of physical objects, the simulation can contain any number of virtual objects. Virtual objects with no physical representation are mirrored on the hall floor via the laser projection system. Unity sends simple motion setpoints to the UAVs and receives the resulting positions of the UAVs via the MoCap connection. The low-level control logic is performed onboard the UAVs. UAVs and their trajectories can be simulated at an accelerated time in simulation-only mode.

In this work, TensorFlow is used to develop machine learning algorithms. These algorithms are coupled to the simulation environment via the ML-Agents toolkit [307]. The toolkit implements reinforcement learning on a drone and uses Unity as a training environment. During training, the simulation is executed up to a hundred times faster. The learned behavior in neural networks controls the drone using Unity. The current system uses C# in the Unity simulation, Python and C++ for drone control based on the ROS, and plain C on the embedded systems of the drones.

3.7.6.2 Drone Swarm Setup

The testbed described in the previous section simulates swarm control on a physical drone swarm. The current swarm at the research hall consists of up to 16 drones controlled by an extended version of the open-source project [509]. Based on the architecture in Figure 3.47 a transport scenario in a warehouse is replicated.

The drone swarm flies in a test area indicated by the laser projection system. Humans can enter the test area if they wear laser protection glasses and headbands with markers, so they get recognized by the swarm as a mobile obstacle. Humans can safely move within the swarm as long as they move at moderate speeds. After takeoff, the drones fly in the range between 1.5m and 2.6m in height.

Transport orders are created by the laser when a marked Frisbee disc gets thrown on the ground in the test area. The transport order is generated as a virtual packet projected by the laser system. The target for all orders is an area on the ground indicated by the laser. The drone with the shortest path to the packet flies to it at a low altitude and picks it up. The drone then delivers it to the target area at a low altitude and ascends back to join the swarm. Multiple orders can be picked up by multiple drones simultaneously.

3.7.7 Reinforcement Learning for Micro-UAV Swarm

After successful simulation of the drone swarm in the testbed, we extended the capabilities of the swarm by using machine learning to perform a warehouse task. This work uses Reinforcement Learning (RL) algorithms. In RL, the algorithm is not given examples of optimal outputs but instead discovers them by trial and error [70].

The goal in RL is to find suitable actions for a given situation to maximize the reward. In RL, an agent is an entity that acts based on a policy. The policy defines how an agent behaves based on the observations it perceives at a given time. An agent is embedded within an environment, and at a given instance, it is in a specific state. The value of a given state refers to how rewarding it is to be in that state. From the current state, the agent can take one of the sets of actions that can bring it to a new state, provide a reward, or both. The agent's main objective is to maximize the total cumulative reward that it receives over the long run [664].

Reinforcement learning with drones is being extensively used for various applications such as drone tracking, and following the leader drone in a swarm [14], achieving a decentralized control of a drone swarm [51], collision avoidance [518], and trajectory planning [318]. This work shows that machine learning techniques such as RL can be tested in the testbed environment created in the research hall.

3.7.7.1 Scenario

In this work, the task of transporting an object to the target area is simulated using RL by a single drone as a proof of concept. A drone flying in the testbed is presented with a virtual object. At first, the drone wanders around in the testbed, unsure what to do. Eventually, it picks up the object and delivers it to the target area, getting a reward. After multiple training sessions, the drone learns that picking up and delivering an object is the best way to maximize the reward.

3.7.7.2 Implementation

The ML agent toolkit creates simulated environments using the Unity Editor and interacts with them via a Python API [307]. The toolkit provides the ML-Agents SDK, which contains the necessary functionality to define an environment within the Unity Editor and the core C# scripts to build a learning pipeline. In this work, the task of transporting orders, as described in Section 3.7.6.2, is simulated. The environment is the testbed in the research hall, and the agent is the drone. Initially, the drone needs to learn how much to rotate the motors enabling it to move some specific distance in a particular direction. Then, the drone computes the relative distance between itself and the target to decide what action to take next. The action of picking up the order and delivering it to the target area earns the drone a positive reward. Drone receives a fixed penalty (negative reward) for every other action. The goal is to maximize the rewards and minimize the penalty. Thus, the drone learns to swiftly pick up and deliver an object to avoid a heavy time penalty.

A hierarchical approach integrates the swarm algorithms with RL algorithms. The swarm algorithms such as separation, cohesion, and obstacle avoidance have a higher weightage which ensures collision-free flight and the safety of the persons in the testing area.

An RL technique called Proximal Policy Optimization (PPO) is used in this work. PPO is a family of policy optimization methods that use multiple epochs of stochastic gradient ascent to perform each policy update [587]. PPO uses a neural network to approximate the ideal function that maps an agent's observations to the best action an agent can take in a given state. A simple neural network (developed in TensorFlow) with an input layer, one hidden layer and an output layer is used. The network's output is a vector indicating the direction the drone should fly in. Agents can ask for decisions from the policy either at a fixed or dynamic interval, as defined by the developer. The PPO algorithm is implemented in TensorFlow and runs in a separate Python process. Communication between Python and Unity takes place via a gRPC communication [718] protocol and utilizes protobuf messages.

After training the RL agent for an hour, the trained model is saved. The saved model computes the actions of the drone in the testing phase. While testing, it is observed that the Micro-UAV can perform the transport task successfully using RL. Moreover, a hierarchical approach allows the swarm algorithms to be used with the RL algorithm. The simulations were performed on a single drone, but can be extended to the entire swarm.

3.7.8 Conclusion

This contribution describes a testbed for the development of resource-constrained Micro-UAV swarms. The testbed architecture combines external observation, visualization, and simulation systems with a drone control unit. The architecture in Figure 3.47 successfully integrates external systems with drones and ensures fast data exchange using radio signals. Swarm algorithms formulate a collision-free path for each drone. The Micro-UAVs successfully perform a transportation task using reinforcement learning in combination with swarm algorithms. Therefore, it is possible to integrate machine learning into the current setup, which opens up new opportunities in the use of machine learning with resource-constrained drones. This work describes how drones can be integrated into a process environment and operated in a meaningful way. In addition, an automatic charging station extends the runtime of the swarm.

3.7.9 Future Work

Industrial use of micro-UAVs will increase as sensors become smaller and more efficient for the use on drones. Thus, on-board sensors will be used instead of a MoCAP system for accurate localization in the future. Future works will include autonomous exploration of an environment, which will involve the swarm creating a map from a safe starting point and updating it in subsequent flights. The recharge station described in this paper will form the basis for this work. Moreover, human-machine and machine-machine

interaction will be developed in the hall to create a safe environment for humans to work alongside drones.

Various ML algorithms could be implemented directly on the drones to enable them to perform complex tasks. For example, ML could be used to learn the complex drone trajectories and motion patterns and thus could help to reduce the dependence on some of the systems in Figure 3.49. In the future, the different research halls could be equipped with a 5G installation that allows the drone swarm to fly between halls. Connecting the swarm via 5G to an already existing high-performance cluster for ML will enable larger-scale field studies for further development of the underlying algorithms.

4 Smart City and Traffic

4.1 Inner-City Traffic Flow Prediction with Sparse Sensors

Thomas Liebig

Abstract: The current traffic situation in urban areas and its forecasting are of interest to various application scenarios, as cities become more crowded and jammed. But the observation and monitoring of traffic situations are expensive, and thus estimates need to be imputed for unobserved locations and predicted for future locations.

In this contribution, we focus on a situation-aware routing use case, which prevents traffic jams. This system needs to take into account real-time estimates of unobserved and future traffic and present several probabilistic approaches to estimating traffic quantities.

4.1.1 Introduction: Problem Understanding

Traffic congestions are crucial problems of urban traffic, both for logistics, and passenger traffic. Data-driven, dynamic control and a mobility shift to automated vehicles could potentially ease current problems and lead to a mobility change in urban environments.

However, traffic systems are complex real-time systems with multiple actors, so control is difficult. Moreover, the observation of this process by measurements is sparse and prone to local (spatio-temporal) validity. Several real-time imputation and prediction steps are therefore prerequired, before the computation of meaningful dynamic recommendations for control is possible.

However, if individual navigation could take the predictions of future traffic situations into account, one would be able to avoid congested road segments or to decide on the best mode of travel in advance. Moreover, since some hazards such as traffic jams often occur by excessively high traffic densities, situation-aware trip planning would also cause fewer traffic congestions, and the infrastructure could be used more efficiently.

The tasks posed by situation-dependent routing are

- the prediction of future traffic situations from sparse observations,
- the utilization of dynamic predictions in planning, and
- the evaluation and selection of individual actions.

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.
https://doi.org/10.1515/9783110785982-004

In the following sections, we address these tasks one after the other and reflect the contributions we made to these questions within the CRC 876. The contribution concludes with a discussion that highlights future research directions.

4.1.2 Traffic Prediction from Sparse Observations

In data mining, learning a prediction model is a well-defined supervised learning task. Given labeled observations x, y from a space $X \times Y$, we train a model $f : X \to Y$ such that the expected loss l between prediction $f(x)$ and the truly observed label y becomes minimal. With this model, the prediction in a certain situation can be obtained by the application of the model to new data $y = f(x)$.

However, the modeling process is highly dependent on the available data, and the suitability of this approach depends on the entanglement of the modeled process.

In traffic control, we have a dynamic spatial process, and observed data is valid only for a limited extent in space and time. Moreover, the basic assumption of the supervised learning approach that the process repeats and, therefore, that past observations are suitable to project the future, does not necessarily hold in general. Traffic control and individual decisions depending on the expectation of future traffic behavior are counter indicators of such assumptions.

Based on these difficulties, various approaches to model traffic exist for different model assumptions, modes of transportation, and granularities.

4.1.2.1 Gas Kinetic Models

In contrast to the data-driven supervised learning approach, one could start by observing patterns and physical properties of traffic and representing these in models. This model-driven approach is subject to the physics of transport and traffic theory. One of the basic observable properties of traffic is that individual moving objects (cars, pedestrians, etc.) do not disappear; rather, over time the number of objects entering spatial regions equals the number of objects leaving this region.[1] By formulating of this observation in a conservation law and deriving a general description model, we obtain so-called gas-kinetic traffic models. Systems of differential equations model traffic similar to any liquid or gas. Prominent examples of these models are the *Burgers Turbulence* and the *Navier Stokes Equation*. In macroscopic traffic modeling (focusing on traffic at gross granularity), gas kinetic models are often applied. A possible numeric approximation is *Force Based Models*, which update individual states of particles based on impacting forces and inertia. The critics of these approaches to traffic modeling are manifold [271]:

1 Note that this property requires a sufficiently long observation interval. For example in a living house people tend to rest and stay at night, while in a car park vehicles are stored until departure.

- If vehicles interact, the impulse and the kinetic energy are usually not preserved. Thus, Newton's Third law of motion (actio=reactio) is not applicable.
- The temperature of a vehicle fluid cannot be matched directly, as it is the variance of the vehicle speed.
- Vehicular gases do not move on account of external pressure but are caused by the inner intention to move at a certain speed.
- Due to the various movement targets, separate flows in different directions occur and interact.
- Vehicular behavior is anisotropic.

4.1.2.2 Cellular Automaton

Cellular automata are a widely used model of physical processes. Introduced by Neumann [473], a cellular automaton features discrete space and time, with transition rules defining the future state of a cell based on its previous state and the state of its neighbors. The advantage of cellular automatons over dynamics is their scalability. Accordingly, boundary conditions are often implemented in a cellular automata model because they have a natural interpretation at this level of description (e.g. particles bouncing back on an obstacle). For traffic modeling, the NAGEL-SCHRECKENBERG MODEL is widely used.

A cellular automaton models a Markov chain, and the conditional probabilities of the future state are completely described by the current state. This Markov assumption does not generally hold for traffic at all granularities. Consider the movement on a highway or the queue at a traffic light. Without additional individual information on the past, one could not tell whether a vehicle leaves the highway or continues, whether the queue will start moving, and in which direction the cars turn at traffic lights.

4.1.2.3 Hierarchy of Motion and Dependency Models

Previous models describe traffic as a Markov process. Current traffic observation and a fixed number of past observations suffice to predict the future. This approach neglects the sociological and psychological aspects of traffic. Persons are traveling by purpose and follow a certain plan to achieve this. Hoogendoorn thus defines the hierarchy of motion [280], which represents the different aspects of trip planning.

Observed traffic combines individual traffic plans and thus has a complex dependency structure which is hard to capture. Tobler's first law of geography [689] states that in a spatio-temporal process, geographically close observations are more related than distant observations. However, in traffic processes, this does not hold, as individual movement paths often start in a living area, use a larger street, and branch back to a tiny street where the working place is situated. So, the information about driving on a highway might be less informative to predict the goal of a trip than considering the few starting locations of the trajectory.

The dependency structure of the traffic process can be represented in a model as follows. Given any evidence of the presence of a moving object at several locations $x_i, \ldots, x_j$, we may model the likelihood of being at other places $p(x_k, \ldots, x_l | x_i, \ldots, x_j)$. If these dependencies are formulated without any circular dependencies, the joint probability distribution over the presence of a moving object at all locations factorizes as:

$$p(x_1, \ldots, x_n) = \prod_{i=1,\ldots,n} p(x_i | \text{pa}(x_i))$$

where $\text{pa}(x_i)$ are the parents of x_i in the above equation. This directed model is called a *Bayesian network* and can be represented by a *graphical model* (a graph consisting of vertices and connecting directed edges) encoding dependencies as arrows between the random variables (the vertices) $\text{pa}(x_i) \rightarrow x_i$. A Bayesian network consists of a structure, given by the previous equation and the associated conditional probability tables for each variable based on its ancestors.

The dependency model can be trained directly from data by comparing whether the trained dependency model represents the same distribution as the traffic observations using a suitable loss function to compare distributions, for example the Kullback-Leibler-divergence. But due to the vast amount of random variables (one per addressable location, e.g., a street segment), learning the model requires some relaxation and approximation. In [406, 407], we propose an algorithm to learn spatial *Bayesian networks* from traffic observations.

4.1.2.4 Gaussian Processes

By the central limit theorem, we know that observations converge towards a normal distribution when repeated multiple times or given a sufficient observation time. In the case of multidimensional observations (e.g., traffic counts at various locations in a street network) this converges to a stationary multivariate normal distribution. Assuming these traffic observations are generated by a probabilistic process, we can use the knowledge of the joint probability distribution to impute observations for unobserved locations (similar to the previous section). Under the assumptions described above, we may assume the observations were generated by a multivariate Gaussian distribution.

$$P(\mathbf{f} \mid \mathbf{X}) = \mathcal{N}(0, K)$$

The generating process is completely defined by the covariance between the variables. Due to the finite number of traffic observations (e.g., measuring locations at street segments), we can denote their pairwise covariances in a kernel matrix K.[2] When we observe some of these locations, we may impute the value at the other locations using the kernel matrix.

2 For an infinite number of observations, a kernel function had to be applied.

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_u \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \hat{K}_{-u,-u} + \sigma^2 I & \hat{K}_{-u,u} \\ \hat{K}_{u,-u} & \hat{K}_{u,u} \end{bmatrix}\right)$$

The correlations in traffic values imposed by the traffic network can be represented in the kernel K. Assuming that a person randomly moves on the traffic network, she travels in the network from one location to another. The dependency structure generated by these random walks can be captured by the diffusion kernel [336], where L is the combinatorial Laplacian of the adjacency matrix, and λ is a hyperparameter.

$$K = \left[\sum_{m=1}^{\infty} \frac{\lambda^m}{m!} L^m\right]_{ij} = \exp(\lambda L)$$

This kernel models every route choice as equally likely. But knowledge of real trajectories can be used to weight the adjacency matrix, and the correlation model can be detailed further [409].

$$m = \hat{K}_{u,-u}\left(\hat{K}_{-u,-u} + \sigma^2 I\right)^{-1} \mathbf{y}$$
$$\Sigma = \hat{K}_{u,u} - \hat{K}_{u,-u}\left(\hat{K}_{-u,-u} + \sigma^2 I\right)^{-1} \hat{K}_{-u,u}$$

A handy property of predictions made by a Gaussian process is that these predictions are normal distributions and are not just expectations. Thus, we may quantify the uncertainty of the predictions and use them to find near-optimal sensor placement [408].

However, the calculation of predictions requires an expensive matrix inversion step. With some preprocessing, the data can be grouped into nearly independent local chunks and the calculations can be distributed [111].

4.1.2.5 Markov Random Fields

In previously described time-dependent models (fluid dynamics and cellular automata), the assumption is that future states are dependent on previous ones. This property is called the Markov property. If we now consider a field of random variables, and the Markov property holds for these variables, it is called a *Markov Random Field (MRF)*. A Markov random field is similar to a Bayesian network in its representation of dependencies. The difference is that Bayesian networks are directed and acyclic, whereas Markov networks are undirected and may be cyclic. Thus, a Markov network can represent certain dependencies that a Bayesian network cannot (such as cyclic dependencies). By contrast, it may not be able to represent certain dependencies that a Bayesian network can (e.g., induced dependencies). The graph of a Markov random field may be finite or infinite.

Any positive Markov random field can be written as an *exponential family*, such that the full joined distribution can be written as

$$P(X = x) = C \cdot \exp < w_k, f_k >$$

A multivariate normal distribution forms a *Markov random field* with respect to a graph $G = (V, E)$ if a correlation of zero corresponds to missing edges in the graph.

A tailored method to model traffic with *Markov random fields* is the spatio-temporal random field [499], which penalizes complex structures in the learning process by regularization and thus hold a sparse representation of the underlying distribution. The Markov property holds over time: the next observation is completely defined by current sensor readings. An application of this model to traffic is presented in [475].

4.1.2.6 Poisson Dependency Models

Empirical observations of spatial phenomena are often count values. For instance, density is the number of objects in a spatial area and flow is the number of objects passing a location in a given time interval. While previous probabilistic models primarily model categorical data or multivariate normal distributed observations, Poisson models seem to be a natural fit, as count values are neither binary nor continuous but are discrete with a right-skewed distribution over an infinite range [249]. A possible approach to combining graphical modeling with Poisson distributions is to use an ENSEMBLE of POISSON REGRESSION TREES [249], each modeling a conditional Poisson distribution. With this model, the underlying joint distribution is unknown and local distributions might be inconsistent. Thus, Pseudo Gibbs Sampling [265] is required to impute unobserved measurements from given evidence. This algorithm initializes the unobserved variables arbitrarily at random and then updates these values according to the conditional distribution given its parent variables. After a burn-in phase, which is highly dependent on the initial distribution, this algorithm draws samples from the joint distribution.

For the imputation of unobserved traffic values, this model outperforms exponential models in [247]. On a massive dataset, training these dependency models is challenging. In [451], the authors show how dependency networks can be trained on core sets, a compressed dataset that can be used as a proxy for the original data. For their algorithm, there is a proven guarantee that in the case of Gaussian dependency networks, the size of the coreset is independent of the size of the dataset. This property does not hold in general, i.e., for *Poisson dependency networks*, it does not hold.

4.1.2.7 Conditional Sum-Product Networks

The need for tractable inference in graphical Poisson models led to the adoption of sum-product networks [506] in Poisson distributions [452]. The graphical structure of these sum-product networks consists of a tree having alternating sum and product nodes in the layers and Poisson-distributed random variables in the leaves. For inference, the structure just needs to be traversed once from the bottom to the root. For training the model, independences between sets of random variables are estimated. In case of no independence, similar objects are grouped into clusters, and a sum-node is introduced. In the case of independences, a product-node is constructed.

For modeling traffic, a model that combines the physical properties (as represented by a cellular automaton) with probabilistic data-driven models would be beneficial. In [597, 598], we show how to model Markov processes with sum-product networks by conditioning them on the previous time slice. The model can represent cellular automaton, is data driven, and outperforms previous Poisson dependency networks for traffic prediction.

4.1.2.8 Differentially Private Learning from Label Proportions

Traffic data is usually collected in a centralized manner, which results in high data transfer and data protection risks. It is especially important that data protection risks are addressed by institutions, due to the introduction of GDPR in all EU countries in 2018 [231]. Organizations want to use this data in order to gain more information or predict future sensor states, e.g., "Will the traffic flow stay the same over the next 15–30 minutes?" Accordingly, they also have to be compliant with GDPR.

Therefore we extend the decentralized learning approach from [651, 655] by applying *differential privacy* to label proportions sent between the different decentralized sensor devices resulting in a privacy-preserving algorithm. In general, the `Learning from Label Proportions` (LLP) algorithm stays the same as proposed in [651]. Because it is important to know the structure and flow of the algorithm, we will briefly consider it further. There are m wireless sensor nodes $(n_1, n_2, \dots n_m)$, which store their measurements in $D(i)$ $\forall i \in 1 \dots m$. Each row in $D(i)$ consists of $[t - w, t]$ measurements, where t denotes a timestamp and w is the *window size* of the last w measurements. Each row is assigned a label, which is taken from a measured value from a future timestamp $t + r$. In the first place, those measurements are split into batches $B_1, \dots, B_h$ where $h = \lceil |D(i)|/b \rceil$ and b denotes the size of the batches, in which $D(i)$ will be divided. The *batches* are then used to calculate *label proportions* for each batch. The generated *label proportions* are sent to the closest c neighbors. Each node uses the received *label proportions* to train $c + 1$ models $f_j(k)$, where $k \in 1, \dots, c + 1$ and j is the current node. The prediction is made by doing a majority voting of the $c + 1$ trained models. This approach has the advantage that we can make use of more than only local measured data point while keeping the bandwidth of transferred data low because only *aggregated data* is sent between the nodes. However, privacy cannot be guaranteed by this approach. Assuming we have traffic flow measurement values, with labels 0, 1, 2, 3, 4 and over a time frame of size b only label 4 is present. Then, from the label proportion, it can be inferred that everyone drove that fast during the period.

We solve this issue by applying *differential privacy* to the label proportions. Firstly, we have to calculate the l_1-sensitivity function to know, how much influence a single data point can make on the output of a function $f : D \to R$:

$$\triangle f = \max_{\substack{D', D'' \in D, \\ ||D' - D''||_1 = 1}} ||f(D') - f(D'')||_1 \tag{4.1}$$

For this scenario, D is the current batch B_i, and R is the resulting *label count*. Considering that we have a simple counting query, a single data point can have at most the influence of 1 (see [185] example 3.1). Finally, we can use the *Laplace distribution* to generate noise, which can be added to the *label counts* to be privacy-compliant under ϵ-differential privacy [185]:

$$lap(x, \sigma, \mu) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}} \tag{4.2}$$

$$lap(x, \frac{\triangle f}{\epsilon}, 0) = \frac{\epsilon}{2 \triangle f} e^{-\frac{|x-0|\epsilon}{\triangle f}} \tag{4.3}$$

In the formula above, the position parameter μ is set to 0, and the *scale* parameter is set to $\frac{\triangle f}{\epsilon}$. These parameters have to be set like this to be compliant with the *differential privacy* definition (proof can be found in [185] Theorem 3.6).

The modified algorithm for calculating label counts can be seen below. As mentioned before, the batches B_i are already generated, and possible labels Y are also known. The output $Q(j)$ contains differentially private label proportions for all batches.

Algorithm 3:

Input: $B_1, \ldots, B_h, Y$
Output: $Q(j)$

```
1  Q(j) ← matrix(h, |Y|);
2  for i in 1..h do
3      for j in 1..|Y| do
4          Q(j)_{i,j} ← sum(B_i == Y_j);
5      end
6      // adding noise to label counts
7      m ← sum(Q(j)_i);
8      for j in 1..|Y| do
9          Q(j)_{i,j} ← Q(j)_{i,j} + lap(e = 0, s = 1/ϵ);
10         clip Q(j)_{i,j} to bounds [0.001, m];
11         normalize Q(j)_i;
12     end
13 end
```

Initially, $Q(j)$ is created with dimensions count batches (h) and count possible labels ($|Y|$). Afterward, the label proportions are calculated iteratively for each batch as follows. First, the label counts (see lines 3–5) and the total sum (see line 7) are calculated. Then the Laplace noise, which is calculated by the sensitivity and ϵ, is applied. Afterward, the new value is clipped to the maximum bounds to prevent values that are too large or negative. Finally, the label counts with noise are normalized. The resulting proportion is stored in $Q(j)$.

The proposed approach is based on the existing LLP algorithm to use the decentralized properties and extend this approach by applying differential privacy to the transferred data. This yields reduced data transfer and increased privacy [568].

4.1.3 Efficient Routing with Dynamic Predictions

In a changing world, geo-spatial data is subject to dynamic changes, and geo-information systems are required to incorporate real-time updates in their analysis and computations. In this section, we focus particularly on route-planning systems. While in a static world, many algorithms exist to compute the (shortest) path from a starting location to a target location efficiently (see Section 4.1.2), this problem becomes more difficult in the case of multi-modal trip planning, as with public transport, because temporal constraints, e.g., transit times and departure times, need to be incorporated. In the real world, these static schedules are not met, but delays occur [439], and deviations from the schedule can be observed. The incorporation of this dynamic information in route computation is beneficial, as it provides tractable travel recommendations to the public. The dynamic information on the delays can be achieved by monitoring the positions of the vehicles and by predicting future delays. This enables proactive trip computation.

In this section, we focus on the tractability of dynamic transit computation. Existing single-source shortest path computation algorithms for the dynamic transit problem suffer from their long computation time. `Transfer Pattern`, a very fast route planning algorithm for transit networks, does not guarantee soundness in case of real-time delay information. Our approach [411] overcomes these shortcomings and introduces dynamic transfer patterns, a data structure that encodes which novel transit possibilities are enabled due to the delays.

In comparison with existing dynamic transit-routing schemes in the city of Warsaw, we highlight the performance gain using our method. Our findings are implemented in the commonly used open-source trip planning framework OpenTripPlanner.

Here, we focus on the point-to-point shortest path problem [49], where in a graph $G = (V, E)$ a path between a source $s \in V$ and target $t \in V$ needs to be found such that the cumulative edgewise cost $l(u, v), with(u, v) \in E \subseteq V \times V$ along the path is minimized.

The standard solution to the problem is Dijkstra's algorithm [176]. Given the graph $G = (V, E)$ and $s, t \in V$, it initializes a queue of nodes $Q = V$ and a distance function over $V \times V$ with $dist(s, s) = 0$ and $dist(s, v) = \infty, \forall v \neq s, v \in V$. Until the queue is empty, the node u with the smallest distance $dist(s, u)$ is picked and removed from Q. For each neighboring node of u, the distance is updated as follows: $dist(s, v) := dist(s, u) + l(u, v)$, if the latter is smaller than the former. Dijkstra's algorithm can be sped up by running it simultaneously from both s and t until a common node u is hit. In the slightly modified version of Dijkstra's algorithm A* [258], the order in the priority-

queue for the traversal not only depends on the cumulated costs to reach a vertex in the graph but also on the expected costs to reach the goal from this vertex. Bound by Minkowski's inequality, whereas $||x + y||_p \leq ||x||_p + ||y||_p$ (known as triangle inequality for $p = 2$), A* prunes the search space in comparison with Dijkstra's Algorithm. A sound heuristic for the remaining cost estimation is the geographical distance that is always lower than the road-based distance.

In the case of static cost functions, contraction hierarchies [223] are a data structure that speeds up the A* algorithm and enables trip calculation in large traffic networks. Instead of searching the shortest path directly within the traffic network, contraction hierarchies reduce the search space to the most important connections. In a preprocessing step, these important segments are identified (based on the topology), and the network is extended by edges between these important links.

In contrast to regular road networks, public transportation data enhances a spatial graph with temporal data by adding timetable information. A trip T serves a sequence of stops $stops(T) = (s_1, \ldots, s_n)$, $s_i \in S$. T connects two stops s_a and s_b if and only if $stop(T, s_a) < stop(T, s_b)$. If multiple trips contain the exact same sequence of stops, they form a line [47].

A common approach is to model the dynamic into the graph G and then to apply Dijkstra's algorithm. This results in a time-extended and time-dependent model. In the time-extended model, every transit node is split into multiple vertices for each event (arrival, transit, and departure). The time-dependent model assigns every transit node one vertex, and arcs encode temporal constraints.

A data structure and algorithm, `Transfer Patterns`, introduced by Hannah [48] is considered state of the art in public transport routing. Based on the assumption that during a day, there are only a few optimal routes from stop s to stop t that differ only in the time they take place. In a preprocessing phase, optimal routes are computed as a sequence of transfer stations, neglecting the time component and information about intermediate stations. For each origin and target destination a directed acyclic graph is saved, containing all routes starting with the destination and containing all intermediate stations until the origin is reached.

In a realistic route planning scenario, various delays occur amongst the public transport vehicles. In contrast to vehicular traffic, trams and trains cannot overtake each other, and vehicles in transit networks wait for connections (e.g., connecting trains). This causes delays to propagate differently than vehicular traffic jams. In addition, two modes of transportation may share the same physical resource (e.g., buses or trams riding on a vehicular street). Thus, two forms of delays in transit networks are distinguished in literature: 1) a vehicle is late due to its own reasons, and 2) other vehicles are late caused by the former [462].

Several models for transit delays are reported in the literature. The work in [175] assumes independence. By contrast, [232] allows delays to cumulate. Sophisticated models incorporate dependencies among the vehicles into the delay [276]. In [439], the delays are analyzed visually.

In a trip planning application, real-time predictions of delay are the main benefit, as future delays may influence the route choice. Thus, we highlight two recent works on delay prediction and delay recognition: [217] applies queueing theory and assumes delays to aggregate, and [774] detects delays and unexpected vehicle movement in real time from the GPS traces.

In this work, we do not focus on the prediction but assume that we have information on delays of vehicles (in the commonly used GTFS real-time data format) either from vehicle observations or predictions.

With such dynamics, the trip computation becomes more difficult. Though a previous publication [47] states that transfer patterns are delay robust, this only holds as long as no new transfers are enabled by the delay. In the likely case that novel transfers are enabled, the existing transfer patterns do not represent this information and cannot result in the optimal transit route.

Transfer Patterns were introduced in [48]. The method comprises a data structure and an algorithm for fast transit route computation. In a preprocessing step, all possible connections are pre-computed and stored in a compressed format. For each public transport line, a table is stored, denoting in the columns the stops along the line. In this way, it holds the maximal possible route without changes.

Our approach to the dynamics of the transit information is to incorporate potential delay information already in the pre-computation phase, and add additional transfer possibilities to the DAGs created during transfer pattern creation.

As we aim to apply the transit route computations in an industrial context, we extend the capabilities of the existing open-source platform OpenTripPlanner (OTP). Our dynamic transfer patterns outperform the algorithms previously available in OTP A* [258] and RAPTOR [165] by an order of magnitude [411].

4.1.4 Control and Planning of Individual Actions

Urban areas are increasingly subject to congestions. Most navigation systems and algorithms that avoid these congestions consider drivers independently and can, thus, cause novel congestions at unexpected places. The precomputation of optimal trips (Nash equilibrium) could be a solution to the problem but due to its static nature is of no practical relevance. By contrast, we describe an approach to avoid traffic jams with dynamic self-organizing trip planning.

In [412], we apply reinforcement learning to learn dynamic weights for routing from the decisions and feedback logs of the vehicles. In order to compare the routing regime against others, the validation uses an open simulation environment (LuST) that allows the reproduction of the traffic in Luxembourg for with varying penetration rates. All of these experiments reveal that the performance of the traffic network is increased, and the occurrence of traffic jams is reduced by applying our routing regime.

Traffic closely resembles a bandit feedback learning environment (see [33] for an introduction to bandit learning). Bandit learning is a reinforcement learning task where the behavior of some blackbox (e.g., a bandit) should be learned only by the feedback we observe. Several actions can be taken (in the bandit problem, this equals drawing an arm). However, only the result of the actions can be observed, and it is unknown what would have happened otherwise. Vehicles serve as agents that move in a road network. The actions are represented by the roads a vehicle can choose at an intersection. Once a road is chosen, a reward will be assigned for that particular road depending on its actual state. The reward for all other roads that could have been chosen remains unknown. This lack of fully labeled data makes a supervised learning approach particularly complex.

The *Policy Optimizer for Exponential Models* algorithm (POEM) [666] is able to learn solely based on the reward values provided by the environment. Additionally, POEM does not perform online learning but rather uses logged data. This abstraction is known from bandit problems, which seek to optimize a reward from the sole information gained after turning the arm of the bandit. This presents a more robust approach, since a learned model can be thoroughly tested before deployment. The system will also not evolve over time, which could lead to unpredictable behavior. This is particularly undesirable in the context of vehicle routing.

In [666], POEM assigns a structured output to an arbitrary input based on its probability of being correct. Therefore, before applying POEM to congestion avoidance, a suitable mapping of the routing problem to a policy h_0, along with an input space X and output space Y, must be modeled. Additionally, a cardinal loss feedback mapping δ is required, which serves as the reward function for all selected input/output combinations.

The input space X was chosen as $X := [0, 1]^m$. Here, each $\vec{x} = (x_1, \ldots, x_m)^T \in X$ represents a feature vector of (normalized) sensor measurements for a road segment. For instance, a road's density, occupancy, mean speed, vehicle count, or waiting time can be used. Any value not in $[0, 1]$ was scaled using min-max scaling.

The output space must be a set of suitable, structured outputs. As POEM should be applied to the problem of congestion control, a single label indicating whether a road is congested or not already provides adequate results. Thus, let $Y := \{(0), (1)\}$, where (0) indicates a road is not congested and (1) corresponds to congestion.

The policy $h_0(Y \mid \vec{x})$ is a probability distribution over the output space. In other words, it assigns a probability to each output $\vec{y}$ given any input $\vec{x}$ based on how likely $\vec{y}$ is to be correct under conditions $\vec{x}$. Hence, predictions are made by sampling $\vec{y} \sim h_0(Y \mid \vec{x})$. The goal of POEM is then to improve this policy. Initially, no such policy exists for the constructed input and output spaces. This is a common problem when applying POEM. Therefore, a default policy is used (compare [666]). Let $h_0(\vec{y} \mid \vec{x}) := 0.5$, meaning both labels are assigned a probability of 0.5 for all $\vec{x}$.

Lastly, in order to improve an existing policy, POEM requires a cardinal loss feedback mapping $\delta : X \times Y \to \mathbb{R}$. This was achieved by applying one of the following

two primitive congestion detection methods to the sensor readings: the primitive density congestion metric, $\delta_{density}$, assumed a road to be congested when its density was greater than one-seventh of its jam density [124]. The primitive mean speed congestion metric δ_{speed} would assume a road as congested when its mean speed was less than ten kilometers per hour of its allowed maximum speed.

In order not only congestion but also reduce it, vehicles must receive frequent information updates about the current state of the road network. Then, POEM is used to predict the next state of the road network. This information will be used by vehicles to bypass roads which are deemed congested. Thus, those results must also be applied in a routing algorithm, such as Dijkstra or A*.

Let $G = (V, E, c, q)$ be a graph representing a road network. Here, c and q are the default cost and heuristic functions. Additionally, assume all vehicles have knowledge about a congestion labeling policy $h \in \mathcal{H}_{lin} \cup \{h_0\}$ [666]. When using dynamic routing, vehicles will receive updates about roads at regular intervals $T \in \mathbb{N}$. The update can then be written as $u_T : E \nrightarrow X$.[3] Then, when a vehicle receives update u_T, it is able to predict how likely a road is to be congested during interval $T + 1$ using h.

The described model receives sensor information only about whole road segments, rather than individual lanes, which might be problematic, as congestion does not always arise on every lane equally. That challenging situation is most likely to occur at junctions where each lane will allow a vehicle to go in a different direction. We address this problem by aggregating sensor data for each connected edge pair (for the use of a line graph of G, see [257]). Additionally, the resulting data allows more precise congestion detection as individual turning lanes are separated in the model.

In order to bypass arising congestion, a vehicle must recalculate its route with respect to the newly received update u_T. This is achieved by increasing the weight of an edge that is likely congested:

$$p^0_{(e_1,e_2)} := h((0) \mid 0.5u_T(e_1) + 0.5u_T(e_2)) \tag{4.4}$$

$$c' : E^2 \to \mathbb{R}_+,\ (e_1, e_2) \nrightarrow \frac{c(e_2)}{p^0_{(e_1,e_2)}} \tag{4.5}$$

The denominator shows the previously mentioned aggregation of sensor data. For notational simplicity, c' is defined for all elements of E^2. However, in practice only a subset of E^2 is used where e_1 is incident or equal to e_2.

The function c' calculates the new weight of an edge e_2 depending on the preceding edge that was reached. For instance, a vehicle on edge $e_1 = (u, v)$ would calculate the weight for edge $e_2 = (v, w)$ using $c'(e_1, e_2)$. A vehicle that starts its route on edge e_2 would use $c'(e_2, e_2)$.

Essentially, c' divides the default weight of an edge by its probability of not being congested in interval $T + 1$. This means the weight of an edge will remain almost

3 Here, it is assumed that updates are received equally for all edges.

unchanged when no congestion is expected. The increase will conversely depend on how likely congestion is to arise.

Finally, it was assumed that sensor data updates are available for every road. In real-world road networks, permanently installed sensors are much more scarcely distributed throughout the network. This problem can be partly alleviated by directly implementing sensors in the vehicle (e.g., using navigation applications provided by smartphones, or self-driving cars). However, some roads will still remain uncovered. Here, u_T can map to $\{0\}^m$. For the previously defined features in X (a road's density, occupancy, mean speed, vehicle count, and waiting time), its dimension m would be equal to 5. This will cause h to assign a probability of 0.5 to both labels (as defined by $\mathcal{H}_{lin}$ in [666]). Another solution might be to map u_T to the average of all sensor readings in an interval. Thus, uncovered roads would reflect the average state of a road network.

Logging

For POEM, no interactive control over actions is required, as it was specifically designed to learn using logged data. Hence, with respect to the previously defined setting, POEM requires a dataset:

$$D := \{(\vec{x}_i, \vec{y}_i, \delta_i, p_i) \mid i \in \mathbb{N}_{\leqslant n}\},\ p_i = h(\vec{y}_i \mid \vec{x}_i)\ . \tag{4.6}$$

This dataset will be created during the logging phase. All edges are assigned weights using c', and routes are calculated using an implementation of A*, which produces the shortest routes for any admissible heuristic. Additionally, POEM is initially applied using the default policy h_0, which will scale all weights equally by a factor of two. The scaling will not affect A*, meaning no route changes will occur, which in turn simplifies learning on previously collected data.

The data itself can either be collected by each vehicle or by a centralized authority monitoring each vehicle. For both approaches, a data entry cannot be created before any feedback is available. Thus, intermediate results must be cached.

First, the aggregated feature vector $\vec{x}_i$ is logged. The respective label $\vec{y}_i$ with its corresponding probability p_i are then determined using:

$$\vec{y}_i = \begin{cases} (0),\ h((0) \mid \vec{x}_i) > 0.5 \\ (1),\ h((1) \mid \vec{x}_i) > 0.5 \\ random((0),(1)),\ otherwise \end{cases} \tag{4.7}$$

Here, $random((0),(1))$ means a label is chosen randomly and uniformly distributed. Lastly, the feedback is logged using either $\delta_{density}$ or δ_{speed}. The respective results will inherently depend on the previously chosen label.

The deployment of our self-organizing routing algorithm in an urban area could be done in two ways. One option is to use the data of an existing stationary traffic information system (e.g., a SCATS [324] system) and feed it into a navigation platform

that can be used by the citizens. The other option is to turn vehicles directly into sensors and retrieve segment-wise statistics on travel time, density, and traffic flow directly from the navigation app. In the latter case, one might be worried about individual privacy because mobility statistics are recorded centrally. However recent work [410] provides an approach to protect individual privacy, known as homeomorphic encryption. This approach encrypts the data such that it still allows for the analysis of the crypto-text, but just the result can be decrypted. In the following, we will test these two deployment settings using stationary and moving sensors and compare them to Nash equilibrium and uninformed routing.

For the comparability of experiments with different routing algorithms, it is essential to guarantee the same traffic demand (i.e., origin/destination pairs) over time. For repeatability of the same origin/destination setting, we perform analysis with a microscopic traffic simulator called SUMO [349]. The simulator models individual vehicles on a microscopic level, so it controls also acceleration and deceleration, and is largely applied in traffic simulation and applications. It allows us to control traffic demand and provides us with complete knowledge of the performance of the street network and the routing performance. In contrast to arbitrary toy experiments, we aim at modeling sound traffic scenarios. We use an open simulation scenario in the city of Luxembourg [143], which enables the reproduction of 24 hours in the city's mobility.

The common procedure of SUMO is to generate the route of each vehicle before the simulation starts, which is why its live routing capabilities are rather limited. However, SUMO provides the *Traffic Control Interface* (TraCI), a network interface that allows full control over the current simulation. We used this to implement a Java application (SUMO-CA) that simulates a central authority. In order to calculate vehicle routes, SUMO-CA loads a road network and converts it to a directed, weighted multi-graph. When running a simulation, SUMO-CA will receive and parse sensor measurements in regular intervals. This information is utilized to predict the next state of the road network using POEM. Finally, those results are used to update vehicle routes.

Unless stated otherwise, each experiment will start at 7:45 (simulation time) and runs over a period of roughly 35 minutes, or exactly 2048 seconds. The reason why this particular window was chosen is that roads generally are more susceptible to congestion during rush hour. Additionally, a size of 2048 seconds allows rerouting intervals to be easily scaled using a factor of two. Finally, in order to create more realistic jams on arterial roads, SUMO was set to scale the original demand by a factor of 1.3.

Evaluating vehicle detours is problematic. Neither absolute nor relative differences will adequately represent measured detours. The reasoning behind this is that long routes will allow longer, absolute detours, whereas, short routes will allow longer, relative detours. Hence, a different metric is required. We propose the usage of the weighted relative detour as follows.

Let $y_A, y_B \in \mathbb{R}_+^\star$ be arbitrary measurements of one vehicle when algorithms A and B are applied, respectively. Then weighted relative detour $diff_{rw}$ will then calculate the

relative difference, while weighting it using the absolute difference.

$$diff_{rw}(y_A, y_B) := |y_A - y_B| \star \frac{y_A - y_B}{y_A + y_B} \tag{4.8}$$

Various charts in [412] present the evaluation results. The baseline is an uninformed Uniform Cost Search (UCS), where each road was assigned its static default weight, and every vehicle chooses its path individually by $A^{\star}$. In this case, congestions are likely to appear. In addition, a Nash equilibrium (NASH) is shown as a baseline.

4.1.5 Discussion

Throughout this work, we highlighted various models to estimate traffic predictions under different model assumptions and properties. The models represent different aspects of traffic at various granularities.

As an example, modeling car-to-car or vehicle-to-infrastructure interactions in inner cities requires different spatio-temporal granularity and thus different model assumptions from a macroscopic daily average traffic flow prediction.

In general, traffic is a chaotic system and the commonly applied Markov assumption is often violated in practice. Future traffic does not only depend on a fixed number of previous observations. Consider, for example, a semaphore in traffic system (a traffic light, a barrier, or a large public parking). In these situations, it is easy to see that, though following certain patterns, traffic is chaotic.

In Google Maps, the inherent structure of traffic data is currently modeled by Graph Neuronal Networks [172, 546]. However p-adic models are also a promising technique at fine granularities to represent the chaotic behavior. Since it is important for production-ready systems that dynamic predictions are tractable, condiditonal sum-product networks [597] are also an interesting future research direction.

We also applied the algorithm used for self-organizing control of navigation plans to control the charging prizes of electric mobility [543]. In this application, we observed that the explored states of the system might be bad for the system provider. As an example, in our experiments, reduced and even negative energy prizes could provide a useful incentive to prevent grid burnouts. However, the total financial risk needs to be bounded. Such constraints could be incorporated into reinforcement learning using stabilities.

4.2 Privacy-Preserving Detection of Persons and Classification of Vehicle Flows

Marcus Haferkamp
Benjamin Sliwa
Christian Wietfeld

Abstract: In some places, the continuously increasing road traffic will soon exhaust the capacity of existing traffic infrastructure unless appropriate measures are taken. Especially in urban environments with a high density of residential and commercial properties, the infrastructure is highly utilized or overloaded during peak hours. Since structural measures are often not possible or only at great expense, a practical solution to counter this issue is to optimize the infrastructure utilization and the control of traffic flows. For this purpose, the widely installed Internet of Things (IoT)-powered Intelligent Traffic Systems (ITS) can be used, which enable automated detection and high-precision classification of different road users and thus transform the infrastructure into a data-driven Cyber-Physical System (CPS).

Although various sensor systems have been proposed, they fulfill only subsets of the requirements, including accuracy, cost-efficiency, privacy preservation, and robustness. One approach that meets those requirements is a novel radio-based sensor system, of which we present two variants in this contribution. The system's fundamental idea is to exploit radio-based fingerprints of road users—multi-dimensional and characteristic attenuation patterns of several radio links—for detection and classification. One of the presented system variants additionally evaluates high-precision channel information extracted from Wireless LAN (WLAN) Channel State Information (CSI) or Ultra-Wideband (UWB) Channel Impulse Response (CIR) data. The proposed solution benefits from increased robustness against a wide range of interferences, e. g., poor visibility due to bad weather conditions. Moreover, the system exclusively uses embedded microcontroller units (MCUs) and radio technologies, allowing compact and cost-efficient installations in rural and dense downtown areas.

We have performed comprehensive field measurement campaigns and machine learning-enabled analyses that confirm the presented approach's high suitability for different requirements and application scenarios. In this regard, we have evaluated multiple applications, including the comparatively simple detection of road users and the fine-grained classifications of several vehicle classes. For instance, the proposed systems achieve more than 99 % for binary classification and 93.83 % for differentiating seven vehicle types.

4.2.1 Introduction

Following the current trend, it is expected that a large part of the existing transport systems will reach their capacity limits in the (near) future. Possible reasons include the approval of new types of personal transport (e. g., e-scooters) and the shift to alternative means of transportation. There are essentially two options to counter this problem without stricter regulatory measures for road users: structural measures to expand current capacities and the efficient utilization of existing infrastructures through optimized traffic flow control. However, the former step is often not an option due to high financial costs and additional long-term restrictions caused by the construction measures. Instead, more efficient traffic flow control is a realistic undertaking thanks to sensor information provided by the vehicles themselves and/or to low-cost IoT components, especially in smart cities. Such systems also collect high-precision and vehicle type-specific information, paving the way for novel and more advanced optimization methods (e. g., type-specific lane assignment or routing and smart parking). For this purpose, the systems must always provide up-to-date and area-wide precise traffic information, which is collected, among other ways, by a sensor network installed over a large area. Next to high accuracy, these systems must also meet other requirements. They should function reliably in challenging weather and traffic conditions while protecting road users' privacy and be energy- and cost-efficient to operate. In some countries, compliance with all these points is a prerequisite for being approved for large-scale installations in road traffic. For instance, some sensor solutions are unsuitable for this use case because of their characteristic weaknesses—e. g., privacy concerns when using camera-based sensors. An increasing number of vehicles is also equipped with GNSS (Global Navigation Satellite Systems) and mobile network connectivity, which provide detailed information about the current position of a vehicle in real time. In the work of Niehöfer et. al. (for example [476, 477]) it has been shown within the CRC that the accuracy of the vehicle position can be enhanced through in-depth system simulation to provide lane-specific positioning information of vehicles. Yet, any system that collects such location information about individual vehicle tracks raises privacy concerns.

Therefore, this contribution presents a novel Wireless Sensor Network (WSN) for detecting and classifying different types of road users, which identifies those based on characteristic inferences of the signal strength of a radio signal (fingerprint). Initially, the wrong-way driver warning system [250] has leveraged the idea of inferring the travel direction of passing vehicles based on the time sequence of the radio links' attenuation. We have enhanced this approach to determine vehicles of certain types utilizing class-specific fingerprints induced by their shapes and materials. We use supervised learning techniques to extract such class-specific similarities from the channel information for the evaluation. Specifically, in this contribution, we present two generations of the novel detection and classification system. The focus here is on the first generation, which correlates the Received Signal Strength Indicator (RSSI) of

many diagonal and transverse radio links. It can reliably infer different vehicle types based on this information [252, 621].

Furthermore, we present a modular and more compact system design that adds further high-precision channel information using WLAN CSI and UWB CIR radio technologies in addition to coarser information [251]. Since our research's focus has primarily been in the context of the initial system design, this section is devoted to the original system. Also, it provides a brief outlook of the successor system.

Figure 4.1 shows the presented IoT-powered sensor system's intended information flow and its use in a smart city context. Here, the communication modules acting as sensors record fingerprints of passing vehicles and preprocess this raw data and the classification task. One could use such exact traffic information in two different application scenarios. In *on-site* applications (e. g., parking-lot balancing, wrong-way driver detection), the acquired data is evaluated immediately on-site and serves as a trigger for further actions (e. g., warnings of wrong-way drivers). By contrast, *global* applications aggregate this locally relevant information to enable predictions and recommendations for larger areas. Finally, the widely deployed sensor systems can dynamically adjust their predictions by periodically verifying the prevailing traffic situation.

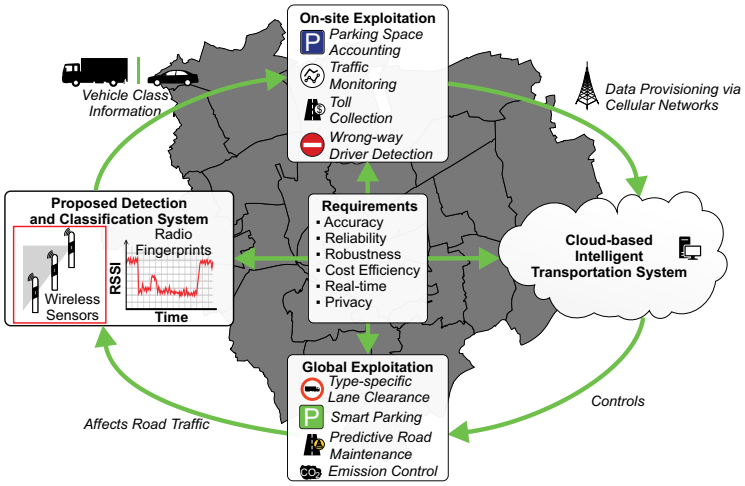


Fig. 4.1: Overall system vision: Embedding of the proposed IoT-based sensor system in a smart city environment. All sensor deployments are *locally* exploited for on-site applications and contribute their data to the *global* data-driven ITS applications ©[2020] IEEE. Reprinted, with permission, from [634].

4.2.2 Related Work

In this section, we provide an overview of existing systems and technologies for vehicle detection and classification. Figure 4.2 shows the abstract process flow and the main logical components of such systems, starting with the gathering of vehicle traces up to the final classification task. The sensor technology generates accurate traces as continuous and high-rate data streams, of which only a part is relevant for the subsequent process steps. To determine suitable sequences within these traces, a detection stage is typically connected afterward, reducing the overall workload. Based on these selected sequences, relevant (statistical) features are then extracted, which serve as input for supervised classification procedures using well-defined taxonomies.

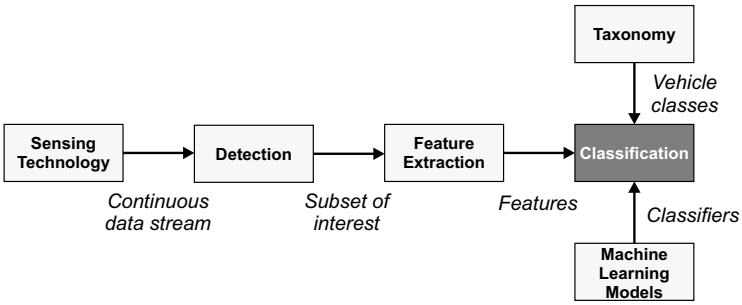


Fig. 4.2: Abstract system model of vehicle classification systems, ranging from data acquisition using different sensor technologies to the final classification tasks ©[2020] IEEE. Reprinted, with permission, from [634].

4.2.2.1 Taxonomies for Classification of Vehicles

The Federal Highway Administration (FHWA) proposes a 13-class scheme for classifying vehicles mainly based on the number of axles [327]. However, this approach's disadvantage is that the number of axles does not indicate the vehicles' exact dimensions. For example, accurate vehicle length information is essential for providing reliable parking space balancing or parking guidance systems.

An alternative taxonomy is the Nordic System for Intelligent Classification of vehicles (NorSIKT) [698], which is used in Scandinavian countries and, with its hierarchical approach, provides different gradations.

The *ISO 3833-1977* standard, the *2007/46/EC* Directive of the European Parliament, and the European New Car Assessment Programme (*Euro NCAP*) provide different schemes for classifying vehicles.

Nonetheless, many academic approaches use individual classification schemes to evaluate the performance of the proposed systems. Following this example, we have developed an adapted scheme with different refinement degrees in this work (cf. Section 4.2.4.1) and applied it to the final performance evaluation.

4.2.2.2 Sensor Technologies Used in Vehicle Detection and Classification Systems

This section provides an overview of established sensor technology used in vehicle classification detection and classification systems. Although much of the referenced work provide performance evaluation in terms of typical metrics—mainly classification accuracy—we want to note that comparing different solutions is difficult. Reasons for this include diverging taxonomies, different approaches for data preparation, and a variety of ML methods—e. g., Support Vector Machine (SVM) [147], Random Forest (RF) [102], k-Nearest-Neighbor (kNN), or Artificial Neural Network (ANN) [236]—used for analysis in the respective works. The sensor systems used can be broadly classified as *intrusive* or *non-intrusive*, respectively.

Intrusive systems represent the *classic* solution approach and are typically embedded in the road surface gathering technology-specific measured parameters. The used sensor technology directly affects the type and scope of measures required for installation or maintenance. While a minimally invasive cutting of the road surface is sufficient for some systems, more extensive and costly measures are necessary for other approaches. Representatives of this system category include Weigh in Motion (WIM) systems, inductive loop detector (ILD)—using one [145] or more inductive loops [365]—fiber Bragg grating sensors [674], vibration sensors [747], and piezoelectric sensors [519].

Non-intrusive systems include acoustic sensor systems, inertial sensors, camera-based approaches, and radio-based solutions. *Acoustic sensor systems* identify road users based on the emitted sounds. The fundamental challenge for these systems is the extraction of the relevant signal component from the dominant noise caused by the traffic noise. However, studies of acoustic sensor systems have shown that their use is of limited value due to comparatively low classification accuracies [225]. By adding other sensor technologies, the precision of these systems is significantly increased [157].

Different types of *inertial sensors*, such as accelerometers, gyroscopes, or magnetometers, are often combined on an inertial measurement unit (IMU). For vehicle detection and classification, the IMUs are either installed directly on the road's surface or at its side. One approach is detecting the number of axles of a passing vehicle, from which the vehicle class is deduced. For example, such systems achieve accuracies of 98.98 % for detection and 97 % for length-based classification [40].

Camera-based systems use pattern recognition and image processing techniques and are widely used due to their high precision. Apart from the detection of road users and the classification of vehicle types, the available high resolutions also allow a reliable recognition of vehicle makes [612], which can be problematic due to regulatory requirements to protect the privacy of road users. Most of these photosensitive systems use ambient light, so these systems' performance varies significantly with the day or visibility conditions. Using a Convolutional Neural Network (CNN), the approach presented in [178] achieves an accuracy of 95.7 % in daylight and 88.8 % in darkness for the classification of six types of vehicles.

In contrast to RSSI, the evaluation of WLAN CSI provides frequency-specific channel information. Orthogonal Frequency-Division Multiplexing (OFDM)-based radio technologies such as IEEE 802.11 use this information to estimate a channel's interference—e. g., multipath propagation—and reconstruct the original symbols. Depending on the number of transmitting and receiving antennas and the channel bandwidth, between 64 and 512 subcarriers are sent in a data packet's training fields. The receiving unit can infer the radio channel's interference by comparing amplitude and phase information of the expected and the received subcarrier sequence. Apart from reconstructing the original symbols, a variety of applications can exploit such detailed information. In addition to vehicle classification [732], localization and tracking of people behind walls and doors [9], as well as privacy-preserving monitoring by healthcare applications [316] are possible. Another technology, UWB, is predestined for the precise measurement of a radio channel because of its high robustness against interference due to its support for large channel bandwidths and its ability to determine accurate channel impulse responses. Although the primary use of UWB is in the area of localization—and recently also as an additional security measure for radio keys—it can also be used for activity detection [599] and vehicle detection classification [251]. *Radio-based approaches* assume that different vehicle types, due to specific shapes and installed materials, characteristically attenuate a radio signal. These attenuation patterns—symbolically referred to as fingerprints—can distinguish between different vehicle classes. Several radio technologies such as Bluetooth [58] or IEEE 802.15.4-based variants [250, 252] are suitable for radio-based methods, provided that the transceiver modules allow access to specific indicators of signal strength. A common approach is to use the RSSI, which is a coarse measure for assessing the received signal strength and depends heavily on the Signal-to-Noise Ratio (SNR) of the radio signal. Since these systems operate in the 2.4 GHz radio range, they exhibit high robustness to poor weather conditions due to rain and snowfall [150, 522].

4.2.3 Radio Fingerprinting-Based Vehicle Detection and Classification

This section describes the two variants of the proposed radio-based systems for vehicle detection and classification, including all essential components. Although both systems follow similar approaches with the evaluation of radio fingerprints , there are differences concerning the hardware components and the data processing, which we discuss in separate sections. First, the original system, which evaluates the signal strength information (RSSI) of multiple transverse and diagonal radio links, is discussed in detail (cf. Section 4.2.3.1). Subsequently, we highlight the significant differences and innovations of the current system approach that leverages high-precision WLAN CSI and UWB CIR channel information in Section 4.2.3.2.

4.2.3.1 RSSI-Based Vehicle Detection and Classification

The system setup initially used for the detection of wrong-way drivers [250], consisting of a total of six radio nodes integrated into delineators—three transmitter and three receiver units each—is shown in Figure 4.3. The system setup uses a constant longitudinal spacing of $\Delta_{lon} = 5m$.

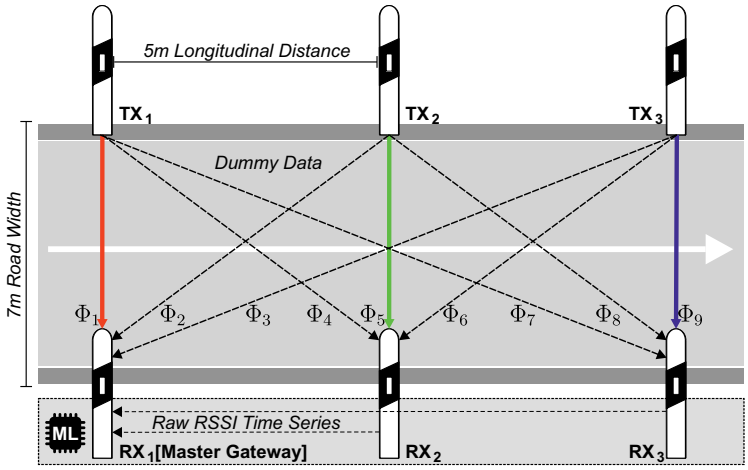


Fig. 4.3: Schematic system overview. Each delineator post contains an RF transceiver module. In total, the system uses nine different radio links ©[2020] IEEE. Reprinted, with permission, from [634].

All nodes use low-cost, off-the-shelf MCU with IEEE 802.15.4 radio modules equipped with omnidirectional antennas and operate with a transmit power of 2.5 dBm in the 2.4 GHz frequency band.

For continuously measuring the RSSI of all radio links, the corresponding transmitter modules periodically transmit pseudo data every 8ms. The system uses a coordinated channel access scheme utilizing tokens to avoid interference between the radio links. Then the receiving nodes send the signal strength information they measure to the master gateway, which aggregates the raw data and synchronizes it for further processing. Figure 4.4 illustrates an example of time-varying radio fingerprints gathered for all radio links for a passing car (top) and a truck (bottom), respectively.

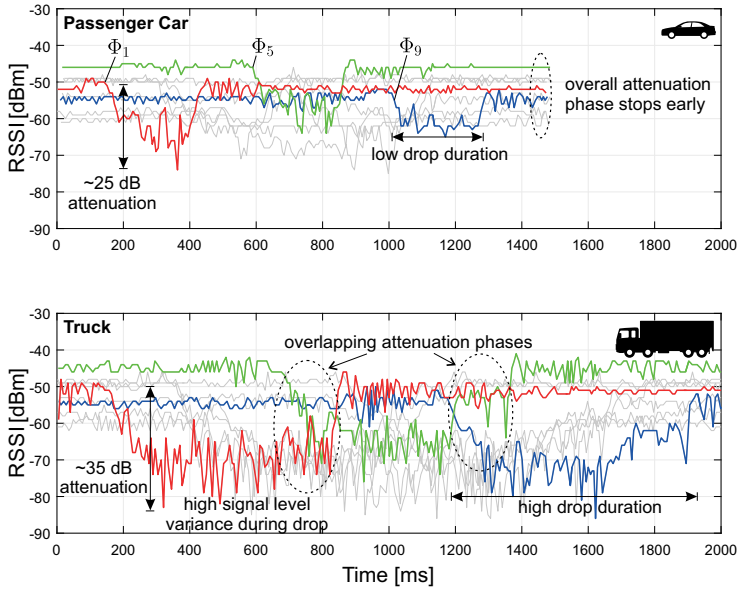


Fig. 4.4: Example multi-dimensional radio fingerprints for a passenger car and a truck. The colorized signals refer to the transverse radio links; the gray signals correspond to diagonal ones ©[2020] IEEE. Reprinted, with permission, from [634].

Figure 4.5 illustrates the entire data processing process for the RSSI-based classification system. First, the RSSI time series of all nine radio links Φ_i are recorded as vehicles pass through. Our approach then forwards the time signals to the data preprocessing block consisting of filtering using a moving average filter and subsequent normalization. These steps are relevant for minimizing the influence of scattered outliers—e. g., multi-path effects—and enabling high compatibility with various machine learning methods (*feature scaling*). Another process block realizes the detection of relevant subsets from the preprocessed time series. The system uses an automated thresholding approach to determine the individual start point t_{start} and endpoint t_{end} for each time series.

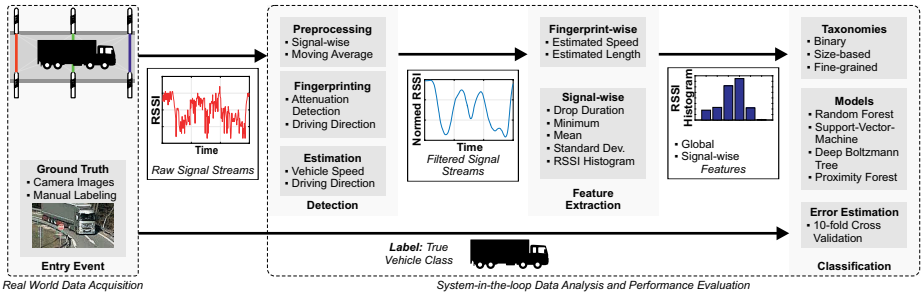


Fig. 4.5: System architecture model and data preprocessing pipeline used for ML-based vehicle classification using radio fingerprints ©[2020] IEEE. Reprinted, with permission, from [634].

The sequences tailored in this way serve as input for the subsequent process steps. The driving speed estimation serves as an additional feature for the classification process. With the help of the known longitudinal distance Δ_{lon} between the individual delineators, the system can estimate the average speed $\tilde{v}$ of the passing vehicles utilizing the temporal difference of the attenuation of the transverse links Φ_1, Φ_5, and Φ_9 using the following equation:

$$\tilde{v} = \frac{1}{3}\left(\frac{d(1,5)}{\Delta t(1,5)} + \frac{d(1,9)}{\Delta t(1,9)} + \frac{d(5,9)}{\Delta t(5,9)}\right) \tag{4.9}$$

where $\Delta t(i,j) = t_{start}(j) - t_{start}(i)$ and $d(i,j)$ is the longitudinal distance between the traversal radio links $\Phi(i)$ and $\Phi(j)$. Negative velocities $\tilde{v} < 0$ refer to an opposite direction, which indicates a wrong-way driver in the case of one-way streets. Similarly, we use Equation 4.10 to determine an approximation of the vehicle length:

$$\tilde{l} = \frac{|\tilde{v}|}{3}(\tau(1) + \tau(5) + \tau(9)) \tag{4.10}$$

where $\tau(i) = t_{end}(i) - t_{start}(i)$ denotes the duration of the attenuation of radio link $\Phi(i)$. The system also considers 90 different indicators—ten features for each of the nine radio links. These represent statistical variables such as mean value, standard deviation, minimum or maximum. In this way, a dimensional reduction is performed, since instead of several hundred signal strength values, the system only needs to process ten features per radio link.

4.2.3.2 Using CSI and CIR Data for Vehicle Detection and Classification

Like the previously presented system, the current modular system proposal also relies on the assumption that it is possible to reliably distinguish different road users by analyzing the characteristic interference they induce to a radio channel. Figure 4.6 illustrates the novel system approach's structure, which uses the radio technologies *WiFi 4* (IEEE 802.11n) and *Ultra-Wideband* (IEEE 802.15.4a). In addition to comparatively

coarse signal strength information, these technologies also measure a wealth of exact channel parameters.

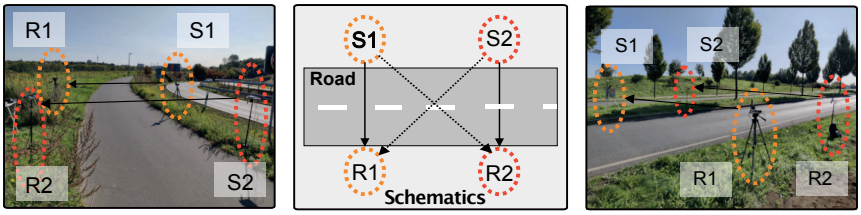


Fig. 4.6: Schematic system overview of the novel system approach leveraging WLAN CSI and UWB CIR data for bicycle detection (left) and motorized vehicle classification (right).

For measuring WLAN CSI, the system uses MCUs based on *Espressif ESP32* with WLAN transceivers connected to directional antennas operating with a transmit power of 20 dBm in the 2.4 GHz frequency band. For a continuous sampling of the radio channel, high-rate dummy packets are exchanged between the respective transmitting and receiving nodes. Each received packet contains CSI information for channel estimation. To reduce protocol overhead and thus increase overall system performance, the system uses unidirectional User Datagram Protocol (UDP) data transmissions. Thanks to an Application Programming Interface (API), the MCUs allow the accessing of CSI and thus amplitude and phase information from various subcarriers. In general, the CSI can contain other fields than Legacy Long Training Field (LLTF) such as High Troughput Long Training Field (HT-LTF) or Space-Time Block Code High Throughput Long Training Field (STBC-HT-LTF), which depends on the supported transmission modes of all WLAN modules involved as well as the channel characteristics. The system currently uses WLAN nodes only for high-rate measurement; the data preparation and ML steps have so far only been performed on more powerful computers.

UWB can accurately determine a radio channel's channel impulse responses thanks to very short signal pulses, allowing further insights regarding a radio channel, e. g., whether a line-of-sight (LOS) path is available or how many significant signal paths exist. The presented system setup uses a custom-made Printed Circuit Board (PCB), combining a *Decawave DWM1000* UWB transceiver module and an *ARM Cortex M3* MCU [687]. Like the WLAN nodes, the systems currently uses the UWB nodes to measure channel impulse responses. This high-resolution channel data is continuously transferred to computers for further processing via USB. Figure 4.7 demonstrates example WLAN CSI and UWB CIR traces.

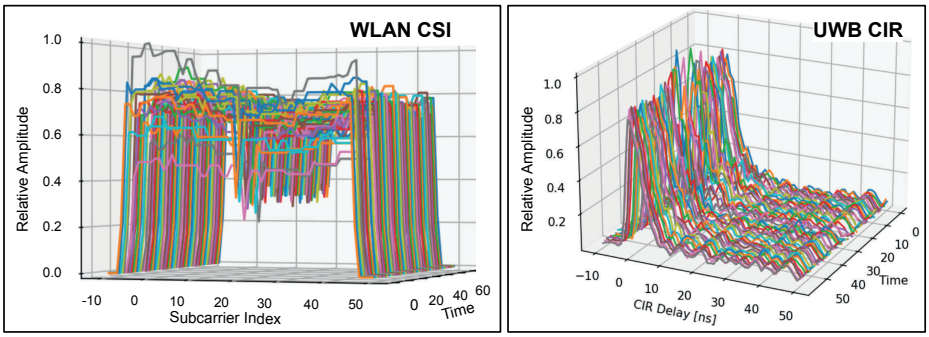


Fig. 4.7: Example WLAN CSI and UWB CIR traces. Each colorized line indicates a complete measurement sample including either multiple subcarriers (WLAN CSI) or CIR buffer sample data (UWB CIR).

4.2.4 Evaluation Methodology

This section presents the methodology used to evaluate both system variants. In this respect, we explain the system setups used for the field measurements, including essential parameters, the taxonomies adopted for the classification, and the ML models for performance evaluation.

4.2.4.1 Field Measurements

For data acquisition, we installed live systems in different environments for both system variants. The original RSSI-based classification system was installed and tested at a rest area on the A9 Autobahn as part of an official test site of the German Federal Ministry of Transport and Digital Infrastructure (shown in Figure 4.8, right). In total, the traces of 2605 vehicles were recorded and then manually labeled using camera images. The main parameters of the RSSI-based system can be found in Table 4.1.

The novel system proposal, which also uses high-resolution WLAN CSI and UWB CIR channel data, was tested at two locations. Traces of cyclists were recorded at a cycle path (Figure 4.8, left), while those of motorized vehicles, especially those similar to passenger cars, were collected at a busy single-lane road (Figure 4.8, right). Thus, the latter setting is similar to that used for the evaluation of the RSSI-based predecessor system. Table 4.1 lists the essential parameters for the novel system proposal.

Tab. 4.1: System parameters of the original RSSI-based system approach and the system evolution using WLAN CSI and UWB CIR.

Parameter	Radio Technology		
	WLAN CSI	UWB	RSSI
Transmission power	20 dBm	-14.31 dBm	2.5 dBm
Operating frequency	2.4 GHz	6.5 GHz	2.4 GHz
Sampling frequency	80 Hz	40 Hz	125 Hz
Antenna type	directional	omnidirectional	omnidirectional
Antenna gain	5-7 dBi	—	—
Number of radio links	1	1	9

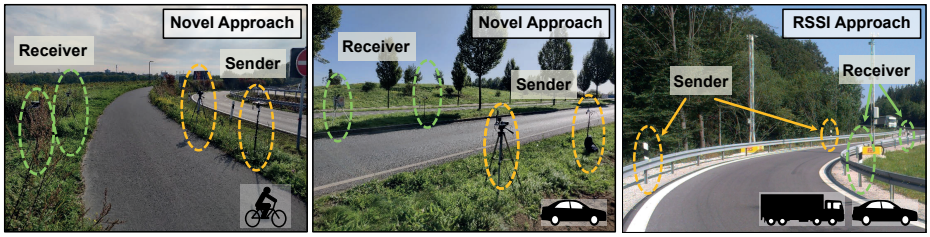


Fig. 4.8: Experimental live deployments of the original RSSI-based system approach (right) and the novel CSI- and CIR-based system evolution (left, middle) on three different settings for gathering real-world vehicle traces.

We used multiple taxonomies for the ML-based performance evaluation of the presented system variants, illustrated by Figure 4.9 for both the original RSSI-based (left) and the novel classification system (right). Defining different taxonomies was necessary because we tested both systems at diverse locations characterized by divergent traffic flows. Specifically, we evaluated the performance of the systems using taxonomies of varying complexity, which we briefly explain in the following:

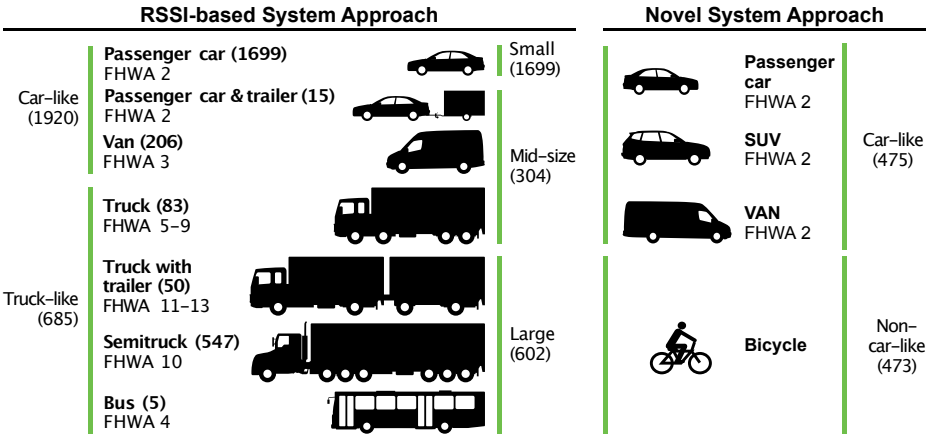


Fig. 4.9: Overview of the vehicle classes and sample counts for the different taxonomies used for evaluating the RSSI-based system approach (left) and the novel system design (right) ©[2020] IEEE. Reprinted, with permission, from [634].

Binary This category distinguishes between *car-like* and *truck-like* or *non-car-like* subclasses. While we classified car-like and truck-like vehicles for the RSSI-based system, we investigated the detection accuracy of the novel system with regard to cyclists using a binary classification with traces of cyclists as well as LOS (*idle*). No object was in the system during the LOS measurements, so fingerprints of the LOS radio channel were measured.

Cyclist vs. Motorized Vehicles Because the dataset of traces for different road users was not large enough, we performed the detection and classification of three classes: *car-like*, bicycle (*non-car-like*), and *idle*.

Size-based This was a 3-type classification of vehicles by vehicle length (only for the RSSI-based system).

Body style-based Here we use a fine-grained classification of seven vehicle types (only for the RSSI-based system).

In the *body style-based taxonomy*, the fine-grained classification task results in an increased overlap of vehicle classes with similar shapes (e. g., bus and semi-truck), increasing the overall classification inaccuracy. Nevertheless, we considered this complex taxonomy for the performance evaluation to illustrate the RSSI-based system approach's strengths and limitations.

4.2.4.2 Machine Learning-Based Classification

We have used several established and state-of-the-art machine learning models to detect and classify vehicles, which we compare and explain below. The following models were used to evaluate the performance of both system variants:

Artificial Neural Networks (ANNs) Inspired by the human nervous system, ANNs have received keen attention for different scientific applications in the context of deep learning. From an implementation perspective, these models are realized by multiple matrix multiplications directly affecting its resulting memory footprint.

Random Forests (RFs) Typical representatives of ensemble learning methods are RFs, which leverage the fact that most instances are assumed to be correct (*wisdom of the crowd*). Random subsets of features and training data are used for training each tree incorporated in an RF. Thanks to their binary decision-making, RFs allow for a resource-efficient implementation using simple if/else statements. By adjusting parameters such as limiting the number of allowed trees or the maximum depth for all trees, both processor and memory utilization can be controlled conveniently.

Support Vector Machines (SVMs) SVMs aim to separate data points in a multi-dimensional space through a hyperplane such that for each feature, the members of each class are separated as precisely as possible, which is achieved by minimizing a specific objective function.

In addition, we have used the following ML models for evaluating the performance of the original RSSI-based system approach:

Deep Boltzmann Trees (DBTs) Belonging to deep learning models, DBTs benefit from the fact that users have neither to extract features nor define transformation functions because they automatically derive differentiable functions from the given dataset. As an inherent downside, DBTs also require the user to select proper hyperparameters and a sufficient amount of training data due to the mass of trainable weights.

Proximity Forests (PFs) Like RFs, PFs belong to ensemble learning models, but instead of CART trees, they utilize proximity trees. Proximity trees use associated data points from the training set and implement–as its name suggests–a proximity-based approach where an object follows the branch with the highest similarity regarding a parametrized similarity measure.

4.2.5 Real-World Validation

This section presents and discusses the results for both proposed vehicle classification system approaches. Because we have developed and tested both systems independently—i. e., in different locations with divergent road users—we cover the results in separate sections, starting with the original RSSI-based vehicle detection and classification system.

4.2.5.1 Radio-Based Detection and Classification System

This subsection covers the results gained for the original RSSI-based system approach for vehicle detection and classification. At first, we describe how we have evaluated its detection performance, i.e., how accurately the system can determine a passing road user. To this end, we fed the raw traces of 2605 vehicles into a system-in-the-loop evaluation setup, allowing for flexible parameter tuning of the detection algorithm. Due to its relatively high system complexity in multiple diagonal and cross-radio links, the system also facilitates speed estimation and wrong-way driver detection (see Figure 4.10). We simulated the latter task by virtually inverting the order of the radio links spanned between the different nodes. Accordingly, an estimated negative speed indicates a wrong-way driver. The histogram shows a noticeable distribution for the dataset, implying a sound detection of the driving direction for passing vehicles. Since the number of detected vehicles matches captured vehicle traces, detection accuracy is 100 %. Nonetheless, we want to note that further real-world measurements are necessary to confirm the results of our virtual detection of wrong-way drivers.

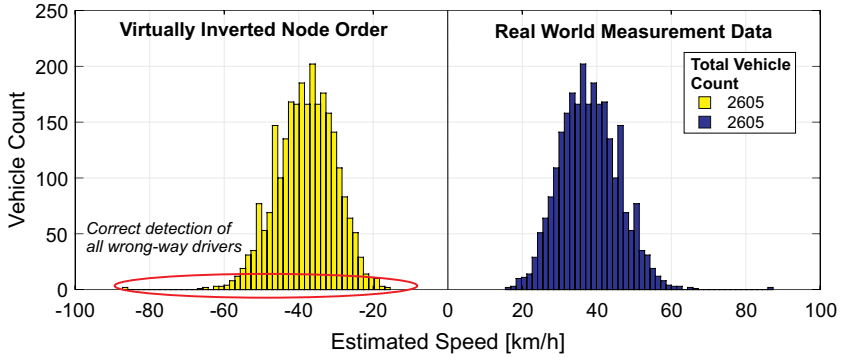


Fig. 4.10: Histograms of the speed estimations for the real world data and the virtually inverted node sequence. The vehicle count matches the total of captured vehicles traces and all wrong-way drivers are detected ©[2020] IEEE. Reprinted, with permission, from [634].

Next, we want to provide and discuss the results of the vehicle classification. We have utilized 10-fold cross-validation with 1/9 data split in each fold, i. e., 90 % of the data is used for training and the remaining 10 % for testing. After ten iterations, the statistical deviations of those folds are derived and used for performance evaluation. Figure 4.11 illustrates the classification accuracies for different machine learning models and the considered vehicle taxonomies (cf. Figure4.9). The 99 % classification accuracy, a typical minimum requirement for some applications, is illustrated as a horizontal dashed line.

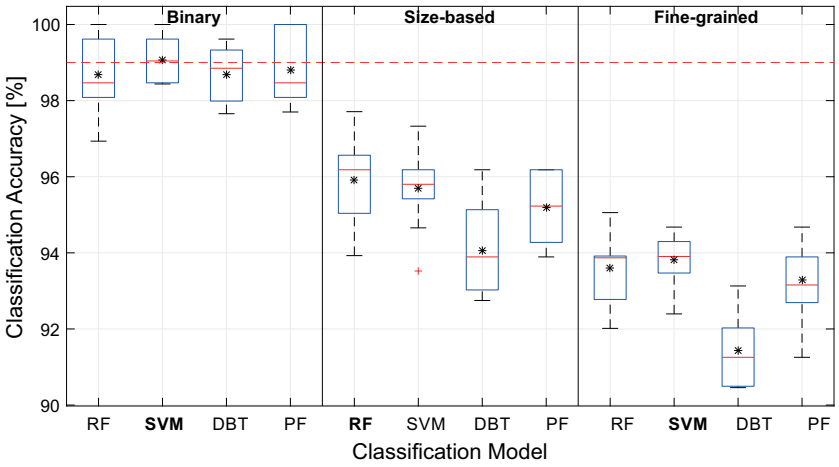


Fig. 4.11: Comparison of the overall classification accuracies for the considered machine learning models and considered vehicle taxonomies ©[2020] IEEE. Reprinted, with permission, from [634].

All evaluated models can exceed the given 99 % threshold for some runs but otherwise fall below it for the binary taxonomy. Only the SVM achieves a mean accuracy that matches this threshold. For the more complex vehicle taxonomies, the overall accuracies of all models decrease significantly: 93 % to 98 % for the size-based taxonomy and 90 % to 95 % concerning the fine-grained task. The apparent deviations between the models' performances result from their different learning strategies. While the stochastic nature of RF induces more significant standard errors in cross-validation, the DBT obtains lower performance levels than the remaining models because it calculates a probability measure for the given data.

Finally, the class-specific classification accuracy for the three considered vehicle taxonomies, i. e., binary, size-based, and fine-grained, is depicted in Figures 4.12, 4.13, 4.14. Starting with the binary taxonomy, which is the most simple classification task differentiating car-like and truck-like vehicles, the main challenge for all models is to classify truck-like cars correctly. We can validate this assumption by interpreting the classification results for mid-sized vehicles, as shown in Figure 4.13: all models have similar standard error values, whereas they perform notably better for small- and large-sized cars. For the fine-grained taxonomy, the multitude of similarly shaped vehicles and the underrepresentation of traces for specific vehicle types (e. g., bus) leads to lower classification accuracies due to larger standard deviations.

4.2.5.2 Vehicle Classification Using WLAN CSI and UWB CIR

For the new modular classification system, we present and discuss the classification results in this subsection. As previously mentioned, we have conducted multiple measurement campaigns for gathering traces of both cyclists and motorized vehicles. Be-

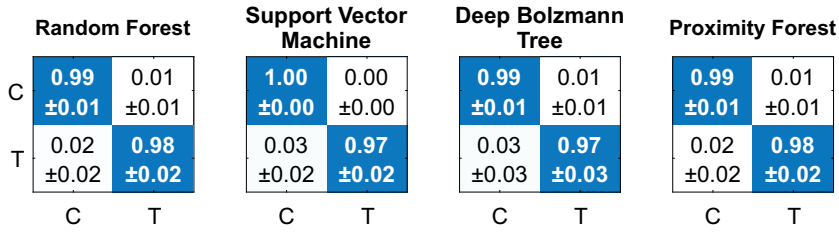


Fig. 4.12: Normalized confusion matrices for binary vehicle classification. *C*: car-like, *T*: truck-like ©[2020] IEEE. Reprinted, with permission, from [634].

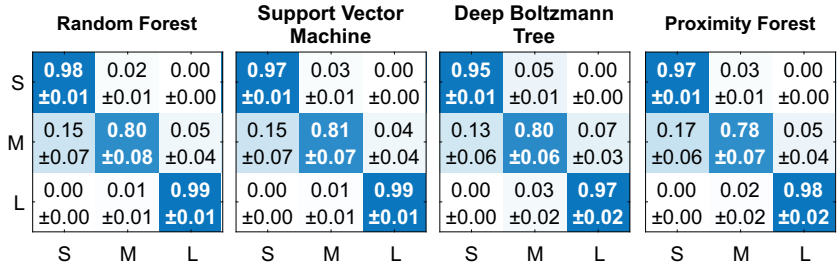


Fig. 4.13: Normalized confusion matrices for size-based vehicle classification. *S*: small, *M*: medium, *L*: large ©[2020] IEEE. Reprinted, with permission, from [634].

cause our measurements have focussed on capturing traces induced by cyclists, we have performed most of the analysis on detecting these. We start discussing the results for cyclist detection. Then we provide the performance results for a multi-type classification task with regard to cyclists and different motorized vehicle types.

For evaluating the bicycle detection performance, we have considered a binary classification task with the classes bicycle and non-bicycle (idle). Table 4.2 lists the maximum classification results achieved for separately analyzing different channel parameters gathered from WLAN CSI and UWB CIR data using ANN, RF, and SVM. Regarding WLAN CSI, the RSSI is the dominant channel indicator leading to the highest classification accuracy for all models. A possible explanation is that the WLAN transceiver modules evaluate multiple channel parameters to extract a significant measure for the link quality. Similarly, there is also a single channel parameter for UWB—the quotient of the estimated first path signal power and the channel impulse response power—leading to the highest classification accuracies. In particular, using this quotient and ANN, we could reach 100 % accuracy for detecting cyclists.

Next, we present the results for the multi-type vehicle classification applied for cyclists and different motorized vehicles. Specifically, we have conducted this task for a total of three classes, i.e., idle, bicycle, and car-like vehicles. Table 4.3 shows the classification results for WLAN CSI and UWB CIR data using ANN, RF, and SVM. Contrary

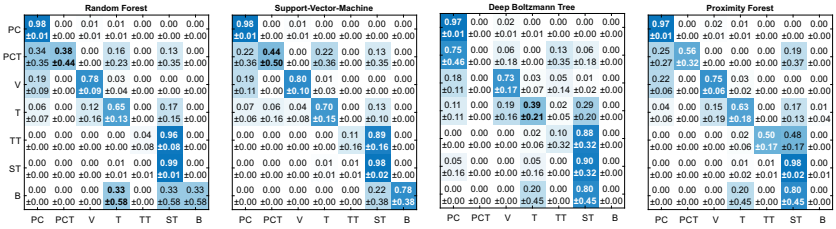


Fig. 4.14: Normalized confusion matrices for body style vehicle classification. *PC*: passenger car, *PCT*: passenger car with trailer, *V*: van, *T*: truck, *TT*: truck with trailer, *ST*: semitruck, *B*: bus ©[2020] IEEE. Reprinted, with permission, from [634].

Tab. 4.2: Bicycle Detection: Results for WLAN CSI and UWB using ANN, RF, and SVM with a 10-fold CV.

Model	Score	WLAN CSI		UWB	
		Value [%]	Param.	Value [%]	Param.
ANN	Accuracy	99.27±0.57	R (f2)	100±0	FC (f0)
	Precision	99.35±0.52	R (f2)	100±0	FC (f0)
	Recall	99.24±0.61	R (f2)	100±0	FC (f0)
	F-Score	99.30±0.56	R (f2)	100±0	FC (f0)
RF	Accuracy	99.45±0.54	R (f0)	99.83±0.26	FC (f1)
	Precision	99.48±0.52	R (f0)	99.84±0.25	FC (f1)
	Recall	99.45±0.51	R (f0)	99.8±0.26	FC (f1)
	F-Score	99.46±0.51	R (f0)	99.83±0.26	FC (f1)
SVM	Accuracy	99.32±0.51	R (f2)	99.83±0.26	FC (f0)
	Precision	99.38±0.47	R (f2)	99.84±0.24	FC (f0)
	Recall	99.30±0.53	R (f2)	99.82±0.27	FC (f0)
	F-Score	99.34±0.50	R (f2)	99.83±0.26	FC (f0)

f: Filter size, *FC*: Ratio of first path signal power and CIR power, *R*: RSSI

to the cyclist detection task, there are multiple predominant channel indicators for each system.

For WLAN CSI, the RSSI seems to be inadequate for achieving the highest accuracy. Instead, the subcarrier's amplitude values of different training fields are more relevant for this task: LLTF when using ANN, STBC-HT-LTF for RF, and several subcarriers in the case of SVM. By contrast, there are two crucial parameters when using UWB: the amplitudes of all raw CIR accumulator data (A) and the amplitudes of accumulator sample 15 (A15). By comparing the resulting classification accuracies for WLAN CSI and UWB CIR data, we can identify a considerable performance gap of about 5 % for the benefit of the former radio technology. However, we note that we have gathered the traces for cyclists and motorized vehicles in different environments with diverging system dimensions, impacting the transmissions between the UWB transceiver modules equipped with internal PCB antennas.

Tab. 4.3: Multi-type vehicle classification: Results for WLAN CSI and UWB using ANN, RF, and SVM with a 10-fold cross-validation.

Model	Score	WLAN CSI		UWB	
		Value [%]	Param.	Value [%]	Param.
ANN	Accuracy	98.23±0.67	L (f4)	92.38±1.30	A (f2)
	Precision	98.52±0.49	L (f5)	93.53±1.46	A (f2)
	Recall	98.31±0.63	L (f4)	93.30±1.34	A (f2)
	F-Score	98.39±0.71	L (f3)	93.41±1.38	A (f2)
RF	Accuracy	98.67±0.62	S (f0)	92.96±1.67	A (f0)
	Precision	98.83±0.59	S (f0)	93.74±1.74	A (f2)
	Recall	98.84±0.60	S (f1)	93.28±1.79	A (f2)
	F-Score	98.8±0.61	S (f0)	93.51±1.75	A (f2)
SVM	Accuracy	96.95±1.66	H_{SC52} (f0)	91.17±2.03	A_{15} (f0)
	Precision	97.86±1.24	H_{SC52} (f0)	92.13±1.85	A_{15} (f0)
	Recall	97.46±0.43	L (f4)	90.48±2.74	A_{15} (f0)
	F-Score	97.39±1.44	H_{SC52} (f0)	91.29±2.25	A_{15} (f0)

A: Amplitudes of all CIR accumulator samples, A_{15}: Amplitudes of CIR accumulator sample 15, f: Filter size, H_{SC52}: HT-LTF sub-carrier 52 amplitudes, L: LLTF sub-carrier amplitudes, S: STBC-HT-LTF sub-carrier amplitudes

4.2.6 Conclusion

This section presented two variants of novel radio-based systems that exploit different indicators of radio channels for accurate vehicle detection and classification. While the original system approach leverages relatively rough attenuation patterns of wireless signals (*RSSI fingerprints*), the evolved modular system approach uses exact channel parameters provided by the radio technologies WLAN CSI and UWB. Compared with existing detection and classification solutions, the proposed system variants are privacy-preserving, robust against challenging weather conditions, accurate, and cost-efficient. We have analyzed the suitability of both systems in comprehensive measurement campaigns in different environments: on a rest area on a highway, a busy one-lane road in a rural setting, and on a cycle path. The presented results approve the high performance of those system approaches for a set of differently challenging applications ranging from simple detection tasks of road users to a fine-grained classification of multiple vehicle types.

In future work, we want to investigate the applicability of different radio technologies (e. g., mmWave) within our detection and classification system. Moreover, we will obtain additional vehicle traces in challenging urban environments (e. g., in a downtown setting) to evaluate and strengthen system performance.

4.2.7 Acknowledgments

In addition to the CRC 876, part of this work has been supported by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) in the course the PuLS project (03EMF0203B).

4.3 Green Networking and Resource Constrained Clients for Smart Cities

Pascal Jörke, Christian Wietfeld

Abstract: The Internet of Things (IoT) will enable a variety of new use cases by linking billions of IoT devices. Introducing new use cases each day, IoT devices will be found everywhere in the future. With a new generation of resource-constrained clients, communication networks have to face new challenges such as high communication ranges, small data transmission efficiency, and large scalability. With the Narrowband Internet of Things (NB-IoT) and enhanced Machine Type Communication (eMTC), cellular communication solutions have been adapted to these new challenges. Including mechanisms for larger communication ranges as well as lower power consumption, NB-IoT and eMTC aim to fulfil the requirements defined by new massive Machine Type Communication (mMTC) use cases. While performance is often only optimized on the lower layers, upper layers including transmission and application protocols need to be addressed by reducing overhead and enabling efficient small data transmissions in order to deliver good performance for resource-constrained clients.

This section describes the achievements in evaluating the performance of Low-Power Wide-Area Network (LPWAN) solutions for resource-constrained clients in terms of energy efficiency, spectral efficiency, and latency. Therefore, new cellular IoT features for power saving and coverage extension are explained in detail, while taking the costs for the scalability of the networks into account. With this knowledge, a performance analysis of resource-constrained LPWAN clients with different coverage conditions is provided.

Although both NB-IoT and eMTC use the same power-saving techniques as well as repetitions to extend the communication range, the analysis reveals a different performance in the context of data size, rate, and coupling loss. While eMTC comes with a 4 % better battery lifetime than NB-IoT when considering 144 dB coupling loss, the NB-IoT battery lifetime has 18 % better performance in 164 dB coupling loss scenarios. The overall analysis shows that in coverage areas with a coupling loss of 155 dB or less, eMTC performs better, but requires much more bandwidth. Taking the spectral efficiency into account, NB-IoT is in all evaluated scenarios the better choice and more suitable for future networks where the number of devices connected is expected to be close to or go beyond the network capacity.

While communication is possible with coupling losses up to 164 dB, the results show that the overall performance is very limited with decreasing signal quality. Although being designed for extended coverage, the mobile network operators should continuously improve the signal quality for both uplink and downlink directions. When increasing

the number of base stations is not feasible, alternative signal quality improvement solutions should be addressed. In this context, the coverage and link quality improvement of cellular IoT networks with multi-operator and multi-link strategies was evaluated as a case study, using the smart city Dortmund, Germany. The results show that the link quality can be improved by up to 13.6 dB, which enables shorter time-on-air for resource-constrained devices and thus drastically improves the energy and spectral efficiency.

4.3.1 Introduction

Waste bins with fill-level sensors, distributed environmental sensors monitoring the overall air quality in large cities, and beehive sensors regulating the hive temperature and transmitting the hive weight are just some use cases that integrate small sensor devices. The IoT enables countless new use cases. While some sensors have fixed power sources, others need to be independent of fixed power sources (e.g. smart waste bins) and therefore must rely on batteries, or, for an even better battery life, energy harvesting [262]. In large-scale scenarios, such as smart waste management, the operational costs need to be as low as possible and therefore the clients have to rely on a single battery for years, but still provide large communication ranges. In the past few years, many new communication solutions have addressed the requirements of low power consumption and wide area communication and are therefore called Low-Power Wide-Area Networks (LPWANs). A promising solutions is Long-Range Wide-Area Network (LoRaWAN), which is used by many public utilities, because it broadcasts in the license-free spectrum and is easy to set up. An alternative in licensed frequency bands is NB-IoT and eMTC, which were derived from the LTE communication technology. Both NB-IoT and eMTC can be deployed in existing LTE networks and therefore provide a fast and easy rollout in many countries.

The next section will give a short overview of the relevant characteristics when considering solutions in the license-free and licensed spectrums.

Clients in the License-Free Spectrum LoRaWAN is an easy-to-use communication solution for IoT. Designed for small data transmissions, LoRaWAN uses a proprietary communication protocol with small overhead and short time-on-air. To further reduce overhead and power consumption, the channel access is based on unslotted Aloha, directly transmitting data when available. With no channel access mechanism and deployment in the license-free spectrum, collisions are inevitable [93], making LoRaWAN unreliable in large scaled networks. Therefore, LoRaWAN is well suited for use cases with minimum Quality of Service (QoS) requirements, where the loss of packets is acceptable. Additionally, duty-cycle limitations need to be taken into account. Duty-cycle limitations are used in license-free spectrum to restrict the maximum transmit time of a device, e.g. 1 %, which means that a device can only transmit data for 36 s each

hour, affecting the maximum transmit interval. More details on clients in license-free spectrum can be found in Section 5.1.

Clients the in Licensed Spectrum When communication needs to be more reliable, communication solutions in licensed spectrum are the better choice. Derived from LTE, NB-IoT and eMTC use central scheduling for available frequency and time resources. Therefore, collisions on the air interface are prevented and the scalability of the network itself is mostly limited by the given frequency bandwidth. The price to pay for a scheduled transmission is the increased overhead for synchronization and control traffic, which affects the spectral and energy efficiency of resource-constrained clients.

4.3.2 Design Objectives of Resource-Constrained IoT Clients

With an exclusive spectrum available, NB-IoT and eMTC (often summarized as cellular IoT) are not limited by duty cycles and can be used with various application protocols and in many use cases. While eMTC relies on an existing LTE network and reuses LTE synchronization, NB-IoT can also be deployed as a stand-alone network [398]. Since NB-IoT uses only 180 kHz bandwidth (compared with 1.08 MHz in eMTC), it can also be deployed in guard bands–the unused bandwidth between two LTE networks–and usually used avoid interference.

Since clients may be distributed over a large area or even in challenging communication environments such as basements, cellular IoT solutions have to provide an extended network coverage, while still enabling low power consumption. The following section will give a short overview on power-saving and range-extending mechanisms that are introduced by cellular IoT.

4.3.2.1 Low Power Consumption

With current cellular communication solutions, the battery lifetime is often limited to a maximum of several weeks. By contrast, cellular IoT has the design objective of 10 years on a single 5 Wh battery [5]. Most IoT devices are designed to transmit small amounts of data on a hourly, daily, or weekly basis, which means the device is mostly in an idle state. Therefore, the new mechanisms for low power consumption focus on energy efficiency in idle mode.

Figure 4.15 depicts a typical NB-IoT transmission cycle. In the connected state, the device transmits its data and waits for a response. When no more data is transmitted or received, the device enters Extended Discontinuous Reception (eDRX). The eDRX mode extends the DRX (Discontinuous Reception) cycle to allow a device to remain longer in a power-saving state between paging occasions [398] and thus to further reduce the power consumption. The device remains synchronized and periodically available for

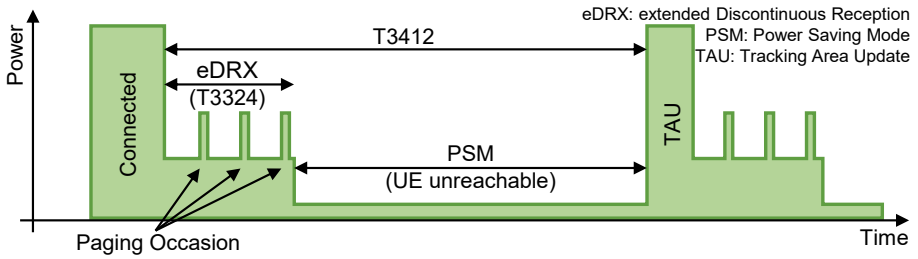


Fig. 4.15: NB-IoT transmission cycle with eDRX and PSM. ©[2018] IEEE. Reprinted, with permission, from [305].

mobile-terminated services. When the eDRX timer T3324 expires, the device will switch into the Power Saving Mode (PSM).

When using PSM, the device enters a power-saving state in which it reduces its power consumption to a bare minimum [398]. In PSM, the device remains registered to the network and maintains its connection configurations. As soon as the device leaves PSM, it does not need to attach to the network; rather, it reestablishes the previous connection, which leads to a reduced signaling overhead and optimized device power consumption. However, the device is unreachable for the network as long as it remains in PSM because it does not listen to the paging time windows. Mobile terminated services have to be suspended until the device reconnects to the network for mobile originated events. Tracking Area Updates (TAU) also trigger the device to end PSM and reestablish the connection to the network. While performing a TAU, the device listens to paging time windows and queued downlink transmissions.

4.3.2.2 Extended Coverage

For a comparism of signal ranges, the Maximum Coupling Loss (MCL) is often used, since it defines the maximum signal attenuation at which the receiver is still able to decode the signal. While the eMTC design objective defines an MCL of 155.7 dB [4], NB-IoT aims to extend the MCL to 164 dB. Figure 4.16 provides corresponding basement penetration ranges for the different MCL objectives.

Besides small bandwidths, cellular IoT solutions use repetitions for an increased energy per bit and therefore improved signal decoding. Therefore, eMTC introduces Coverage Enhancement (CE) Modes A and B. CE Mode A is mandatory and supports up to 32 repetitions while CE Mode B is optional and defines up to 2048 repetitions. NB-IoT also supports up to 2048 repetitions, though it does not divide the number of repetitions in different CE Modes, making all repetition options mandatory to all devices. Table 4.4 gives a detailed overview of the maximum number of repetitions for each NB-IoT and eMTC signal [398].

As similar data is transmitted, the application data rate decreases drastically with each repetition and devices consume more power compared with a transmission with-

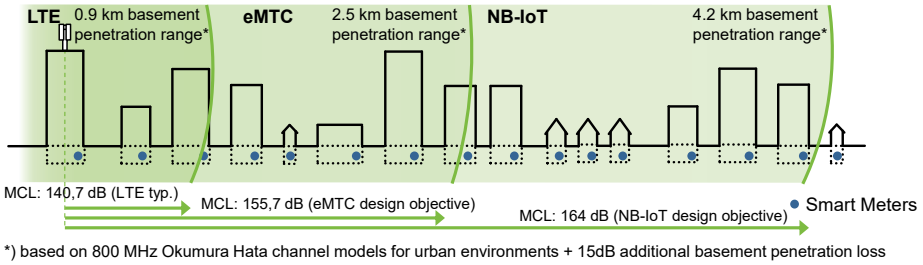


Fig. 4.16: Coverage Enhancement in NB-IoT and eMTC

Tab. 4.4: Maximum number of repetitions in eMTC and NB-IoT.

eMTC	Max. repetitions		NB-IoT	Max. repetitions
	CE Mode A	CE Mode B		
PDSCH	32	2048	NPDSCH	2048
MPDCCH	16	256	NPDCCH	2048
PRACH	32	128	NPRACH	128
PUSCH	32	2048	NPUSCH	128
PUCCH	8	32		

out repetitions. While new mechanisms like eDRX and PSM aim to increase the energy efficiency and battery lifetimes of cellular IoT devices, the extended coverage will lead to a significant reduction of energy efficiency. To ensure that the cellular IoT design objective is still achievable, both NB-IoT and eMTC must be subjected to a performance analysis.

4.3.2.3 Application Protocols for IoT

While LPWAN solutions in a license-free spectrum such as LoRaWAN often lack end-to-end Internet Protocol (IP) support due to the large protocol overhead, both NB-IoT and eMTC are able to transmit IP traffic such Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) messages. Due to the reduced transmission capacity, the packet size and number of message sequences should be as low as possible even with the coverage extension mechanism. For a decent system performance, the choice of a suitable application protocol is essential.

Message Queuing Telemetry Transport (MQTT) is a TCP-based IoT communications protocol, designed for Machine to Machine (M2M) data transmissions in low bandwidth environments [41]. It uses a centralized broker to which clients can publish data, while other clients can subscribe to data updates. In addition to transmission protection through TCP, MQTT introduces three Quality of Service (QoS) levels. QoS level 0 is a simple, low-overhead method of sending a message. The client simply connects to

the broker and publishes the message without MQTT acknowledgement by the broker. The data is still acknowledged on lower layers by TCP. QoS level 1 guarantees that the message is transmitted successfully to the broker. The broker sends an acknowledgement back to the sender. In case of a transmission loss of either the data or the acknowledgement, the data is retransmitted by the sender. QoS level 2 is the highest level of service. It comprises a sequence of four messages between the sender and the broker, and a handshake to confirm that the main message has been sent and that the acknowledgment has been received. When the handshake is completed, both sender and receiver are sure that the message was received exactly once, though this approach creates the highest message overhead.

MQTT for Sensor Networks (MQTT-SN) is an UDP-based, optimized version of the IoT communications protocol MQTT, designed specifically for efficient operation in large, low-power IoT sensor networks [645]. Like MQTT, it uses a centralized broker. Besides QoS levels 0 to 2, a new QoS level is introduced, which further reduces the message overhead. Publishing messages with a QoS level of -1 doesn't require an initial connection setup and broker registration; rather, it only transmits a single publish message with all required data. While QoS -1 comes with the lowest overhead, it does not provide acknowledgements and other responses from the broker.

Constrained Application Protocol (CoAP) is a third important IoT application protocol. CoAP is an UDP-based, specialized web transfer protocol for use with resource-constrained nodes and constrained networks in the IoT [600]. The protocol is designed for M2M applications. Unlike MQTT and MQTT-SN, it transmits data directly between a client and a server. Since no connection needs to be established at first, CoAP comes with a low message overhead. When using confirmed transmissions, all data that is transmitted between server and client is confirmed by application acknowledgements. The data is re-sent until it is acknowledged or the maximum number of retries is reached.

When high QoS is required, MQTT or MQTT-SN should be used for data transmission, since both protocols provide multiple layers of a protected transmission. When energy efficiency is essential and data are transmitted only from point to point, CoAP is the better choice. Figure 4.17 gives an overview of message overheads for these IoT application protocols.

For the highest energy efficiency, the number of transmitted messages should be as low as possible. When no application acknowledgement is required, both CoAP Non-Confirmable and MQTT-SN QoS -1 are applicable. In case of acknowledgement is needed on the application layer, CoAP is the most energy efficient choice since both MQTT and MQTT-SN require a previous connection setup, before transmitting and acknowledging user data.

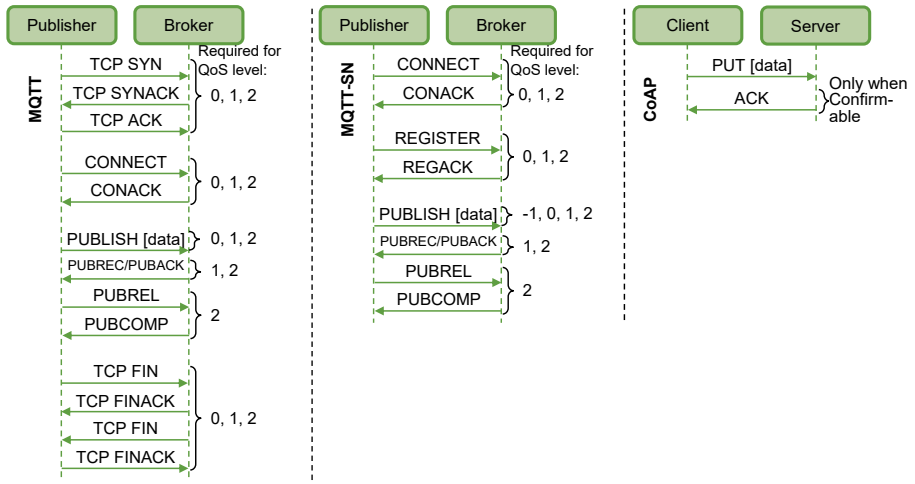


Fig. 4.17: Comparison of message sequences for different IoT application protocols.

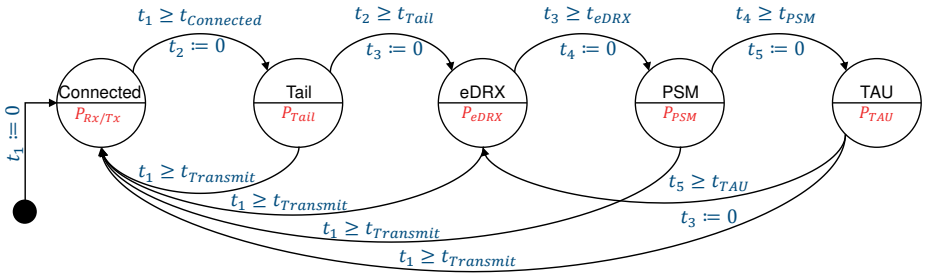
4.3.3 Performance Analysis of Resource-Constrained LPWAN Clients

While features like eDRX and PSM aim for lower power consumption and thus longer battery lifetime, coverage enhancement provides a much higher power consumption for UEs by introducing longer transmission and reception intervals. Both features are required to fulfill the challenges of 10-years battery lifetimes, 164 dB maximum coupling loss, and a maximum latency for a single data transmission of 10 seconds. In this section, performance studies of cellular IoT solutions will be analyzed.

4.3.3.1 Power Consumption Analysis of NB-IoT and eMTC in Challenging Smart-City Environments

In Section 4.3.2.1, two new power-saving states for NB-IoT and eMTC devices are introduced. Both states provide a reduced power consumption compared with current GSM or LTE devices. Besides PSM and eDRX, devices can enter three additional power states: Connected, Tail, and TAU. In the Connected power state, random access, data transmission, and reception are performed. After data transmission and reception, the device remains for a predefined time in a Tail state, also called a data inactivity timer, where it remains connected on the RRC communcation layer for additional data exchange. Then, it switches to the eDRX power state, where the device wakes up only for paging occasions. Finally, the device reduces its power consumption to a bare minimum in the PSM state. It periodically wakes up for TAU and checks if downlink transmissions are queued, since these messages can't be received while in PSM. Figure 4.18 gives an overview of the state machine that is used to determine the energy consumption of UEs.

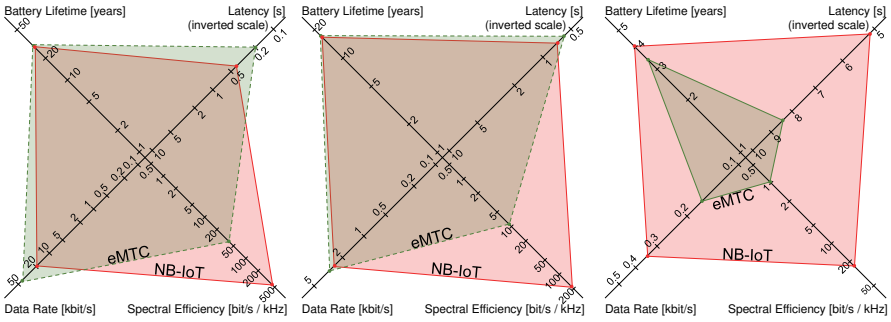
To compare different cellular IoT solutions and assess if all IoT requirements can be fulfilled by NB-IoT and eMTC, the authors in [305] provide a performance analysis of



Connected:	Data transmission or reception	PSM:	Power Saving Mode
Tail:	Idle state before entering eDRX	TAU:	Tracking Area Update
eDRX:	Extended Discontinuous Reception Window	P:	Electrical Power

Fig. 4.18: State machine for power consumption evaluation. ©[2018] IEEE. Reprinted, with permission, from [305].

NB-IoT and eMTC latency, data rate, battery lifetime, and spectral efficiency for three different coverage classes. The results are shown in Figure 4.19.



(a) 144 dB coupling loss **(b)** 154 dB coupling loss **(c)** 164 dB coupling loss

Fig. 4.19: Comparison of NB-IoT and eMTC devices for 84 bytes of acknowledged uplink data every 24 hours in different coverage conditions. Note that the axis scales vary between the three figures. ©[2018] IEEE. Reprinted, with permission, from [305].

The results show that eMTC performs slightly better than NB-IoT at coupling losses of 144 dB and 154 dB. While the eMTC performance gain is rather small, it uses 6 times more bandwidth than NB-IoT, making transmissions less spectral-efficient. When it comes to cell edges such as basements, where the coupling loss can increase to up to 164 dB, NB-IoT clearly outperforms eMTC. While both cellular technologies use the same power saving mechanisms, NB-IoT needs fewer repetitions to transmit data, which reduces the time on air and therefore extends the time in PSM between transmissions.

Although eMTC performs better than NB-IoT in good coverage conditions, the difference in data rate, latency, and battery lifetime performance is rather small. When it comes to poor coverage conditions as well as spectral efficiency, NB-IoT is recommended over eMTC.

4.3.3.2 Coverage and Link Quality Improvement of Cellular IoT Networks with Multi-Operator and Multi-Link Strategies

Section 4.3.3.1 has given an overview of the performance of cellular IoT technologies in different coverage scenarios. With increasing signal attenuation the overall performance decreases drastically. Therefore, the signal quality should always be as good as possible. Instead of expanding the networks by installing new base stations, multi-operator strategies (such as National Roaming) can provide better coverage and link quality for LTE and cellular IoT technologies, by allowing cellular devices to use networks from different Mobile Network Operators (MNOs) as well.

The authors in [306] evaluated the potential of coverage and link quality improvement in terms of multi-operator strategies in the Smart City Dortmund as a case study. By extracting the number and locations of all cellular base stations in Dortmund and applying empirical path loss models for urban environments, we performed a comprehensive coverage analysis. The results are given in Figure 4.20.

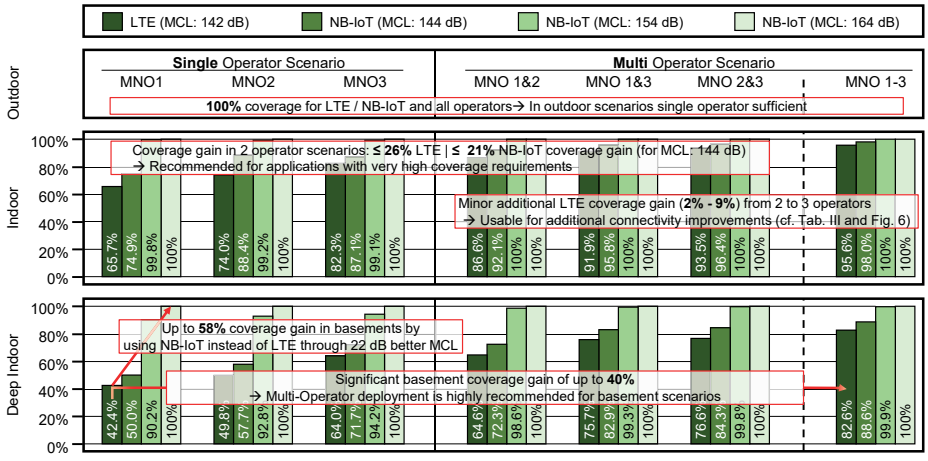


Fig. 4.20: Results of the coverage analysis for outdoor, indoor and deep indoor scenarios and different cellular communication technologies in an urban environment. ©[2019] IEEE. Reprinted, with permission, from [306].

While all MNOs can provide full LTE and NB-IoT coverage outdoors, indoor, and basement coverage, deep indoor coverage from LTE decreases to 66 % and 42 %, respectively.

In multi-operator scenarios, the deep indoor coverage increases by up to 40 %, which makes multi-operator deployment highly recommended in LTE scenarios.

Due to its extended coverage, NB-IoT can provide full coverage in all scenarios when using a maximum coupling loss of 164 dB. If the maximum coupling loss is limited to 154 dB for a better performance (see Section 4.3.3.1), NB-IoT can still provide full coverage when multi-operator strategies are used.

Even in scenarios with full coverage, multi-operator strategies can be reasonable by improving the average link quality by up to 13.6 dB (Figure 4.21). In Figure 4.19, a 10 dB improvement of link quality can already increase the battery lifetime of an NB-IoT device from 4 to 18 years and decrease the latency from 5 s to 0.8 s. Therefore, multi-operator strategies are highly recommended for link-quality improvement.

Average Signal Power Gain for LTE and NB-IoT	MNO 1&2		MNO 1&3		MNO 2&3		MNO 1-3 (National Roaming)		
	MNO 1	MNO 2	MNO 1	MNO 3	MNO 2	MNO 3	MNO 1	MNO 2	MNO 3
Outdoor	9.5 dB	8.0 dB	10.2 dB	8.3 dB	9.0 dB	7.8 dB	11.4 dB	10.0 dB	9.0 dB
Indoor	11.2 dB	9.1 dB	12.3 dB	10.1 dB	10.2 dB	9.0 dB	13.6 dB	11.0 dB	10.6 dB
Deep Indoor	10.4 dB	8.9 dB	11.0 dB	9.1 dB	9.6 dB	8.4 dB	12.6 dB	10.9 dB	10.0 dB

Fig. 4.21: Results of the coupling loss reduction potential for different coupling loss scenarios and cellular communication technologies. ©[2019] IEEE. Reprinted, with permission, from [306].

4.3.4 Conclusion

Energy efficiency is an important factor in the IoT. Many use cases rely on sensors that can last at least 10 years on a single battery. With new communication technologies such as NB-IoT and eMTC, cellular solutions respond to the new challenges that are introduced by the IoT. But optimized communication technologies alone are not sufficient. Energy efficiency needs to be addressed on all layers, from the choice of an appropriate application protocol that produces as low an overhead as possible to link-quality improvement strategies that obviate a high number of repetitions on the air interface. In view of the results of the performance and coverage analysis, a good device placement is of great importance for resource-constrained devices. In the future, extensive measurements of latency and energy consumption can be used to derive an ML-based predictive model for latency and energy performance by using passive measurement parameters such as signal strength and signal quality. Additionally, the number and size of transmitted messages can be reduced by ML-based model-predictive communication, as introduced in [30]. If green networking and resource constraints are taken into account from the very beginning of an application's design, the performance of the system from a user and network perspective can be significantly increased.

4.3.5 Acknowledgments

In addition to the CRC 876, part of this work has been supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK, former Federal Ministry for Economic Affairs and Energy (BMWi)) in the course of the ”Ideal” project, funded under grant agreement no. 03ET7557A.

4.4 Vehicle to Vehicle Communications: Machine Learning-Enabled Predictive Routing

Cedrik Krieger
Benjamin Sliwa
Christian Wietfeld

Abstract: Vehicular communication is used to exchange safety-related status information, enable efficiency-oriented mobility planning, and share other user data, such as video streams, among a locally restricted area and the vehicles and nearby infrastructure within it. Without the need for cellular coverage, the particular devices, or agents, organize themselves in a distributed fashion without a central coordination unit. This ability not only allows the realization of Intelligent Transportation Systems (ITSs) that will have a major impact on the cities of the future, but it also enables spontaneously deployed networks, that cover the task of on-demand network provisioning for events. A well-known example is the support of rescue units that can utilize Unmanned Aerial Vehicles (UAVs) for remote sensing and delegate exploration tasks and reduce the risk of endangering human personnel. These applications all have high requirements and need a robust and reliable communication behavior. As Mobile Ad-hoc NETworks (MANETs) are not managed centrally, data needs to be routed efficiently from the sender to the receiver, whereas link losses and unnecessary hops need to be avoided. Established protocols rely on simple distance measurements and try to minimize the sender-to-receiver distance. In challenging networks, these simple approaches can not cope with the complexity of the task. Therefore, more advanced techniques integrate more information and provide a higher grade of generalization. Comprehensive simulations have shown that the utilization of cross-layer knowledge and the prediction of future network states enable reliable and robust reinforcement learning-based routing algorithms, that achieve high performances under different conditions. Moreover, this technology outperforms established routing protocols by up to 51 % in all considered studies.

4.4.1 Introduction: Direct Agent Communication in Ad-Hoc Networks

Self-organizing networks, where the nodes communicate directly, are described as ad-hoc networks. Routes are built not only to directly reachable neighbors, but also to more distinct nodes with which communication is only possible by invoking (multiple) intermediate hops. This way, all agents create a mesh. Here, the sub-class of Mobile Ad-hoc NETworks (MANETs) is of particular interest, as they explicitly specify the possible movement of nodes, which leads to frequent changes in the network topology and

thus requires suitable solutions for providing a robust and reliable communication. As MANETs are infrastructure-less, efficient routing protocols are required to cover a trade-off between overhead and the provision of valid routes. Whereas too much traffic used for coordination would reduce the overall throughput, outdated information leads to the loss of packets. The nodes' mobility is a primary factor of impact on the network topology. MANETs can have a hybrid consistency of mobile nodes like those of UAVs and cars, but it also has nodes with very low or stationary mobility like those of pedestrians and Road-Side Units (RSUs). Besides mobility, varying channel conditions due to multi-path propagation, especially in urban environments, signal attenuation, and shadowing, have a significant impact on the node's reachability and can harm the end-to-end routing performance. As these varieties of influxes offer challenging conditions for MANETs, established routing protocols that rely on considerably simple metrics, such as distance vectors represented by hop counts, are not able to fulfill the requirement for a robust and reliable communication. Therefore, the integration of further information and the enhancement of routing metrics are motivated in order to assess occurring network situations adequately and increase the overall performance.

4.4.2 Related Work: Evolution of Mobility-Predictive Ad-Hoc Routing Protocols

A classification of the developments of ad-hoc routing protocols is given in Figure 4.22. The proposed protocols originate from the Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.) [472] project located in the Freifunk community. B.A.T.M.A.N. III was originally developed to tackle scalability problems in the established routing protocol Optimized Link State Routing (OLSR) by distributing the topology knowledge among local entities and thus obviate the need to calculate the full network graph on every node, which is an expensive task and especially unsuitable for resource-constrained systems. Subsequent B.A.T.M.A.N. versions relocate their point of operation from Internet Protocol (IP)-based routing in layer 3 to layer 2 in order to provide an network protocol-independent routing approach and have a more direct impact on the packets. As B.A.T.M.A.N. is intended for real-use cases, kernel implementations are available, but, simulation models for scientific research are often omitted. Therefore, a simulation model of the B.A.T.M.A.N V protocol version for the well-known discrete event simulator Objective Modular NETwork testbed in C++ (OMNET++) [701] has been developed and validated by field experiments in [627]. However, as the overall goal of the Freifunk community is to provide mesh-based Internet access within cities, B.A.T.M.A.N. implementations contain overhead to fulfill this task, such as Host Network Announcements (HNAs), is separate from the actual routing process. While the first extension B.A.T.Mobile [623] forks from the main branch and hauls those measures, the consecutive protocol Predictive Ad-hoc Routing fueled by Reinforcement learning and Trajectory knowledge (PARRoT) [635] omits additional overhead for network provisioning. Thus, PARRoT, which gained additional influences of reinforcement learning-based routing,

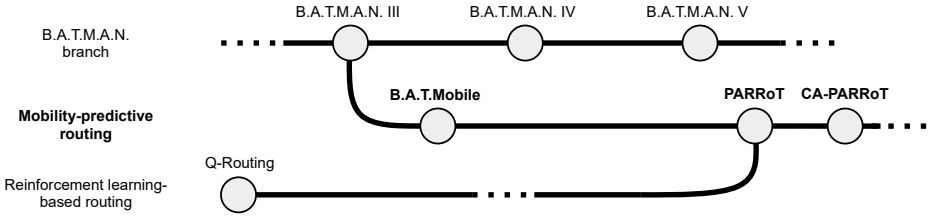


Fig. 4.22: Evolution graph of routing protocols.

concentrates on IP-based routing for an assessment of the concepts. The de-capsulated development of novel approaches intends to merge the newer routing mechanism back into the latest B.A.T.M.A.N. version. The latter protocols, B.A.T.Mobile and PARRoT, follow a mobility-predictive routing approach and are explained in more detail in the following. The most recent development is Context-Adaptive PARRoT (CA-PARRoT) [586], which can be regarded as an extension to PARRoT and follows a hybrid machine learning approach.

Routing protocols can be classified into *reactive* and *proactive* protocols. The first initiate a route-building process on-demand. Well-known examples are Ad-hoc On-demand Distance Vector (AODV) and DYnamic MANET On-demand routing protocol (DYMO). The latter maintain routing tables that are used for lookups when necessary and are updated periodically. Destination-Sequenced Distance Vector (DSDV) and OLSR are widely known proactive protocols. Greedy Perimeter Stateless Routing in wireless networks (GPSR) is a geo-based routing approach that considers mobility and communication as a dependent task. The route building is done by minimizing the geo-distance between sender and destination node with each hop. Extensive summaries about existing protocols are found in [486] and [470]. An empirical analysis of used protocols in vehicular networks is provided in [119].

Recent developments in the machine learning field have also had an impact on routing algorithms. The authors of [671] use a centralized Artificial Neural Network (ANN) to enable a Software Defined Network (SDN) approach for latency minimization in vehicular networks. However, reinforcement learning-based approaches allow routing entities to make autonomous decisions in a decentralized manner. In [99] Q-routing is proposed as an integration of autonomous routing decisions based on learned latency. The authors of [481] extend this with mobility-based metrics and take into account the swarm coherence of agents. A summary of channel and propagation models is given in [704]. The authors of [60] present an approach of learning from stochastic channel parameters. Intelligent routing algorithms and developments for ad-hoc networks might also have an impact on future network generations [17].

4.4.3 Approaches: Enhance Routing by Prediction and Machine Learning

In this subsection, the routing approaches of B.A.T.Mobile, PARRoT, and CA-PARRoT are presented. B.A.T.Mobile leverages cross-layer knowledge from the mobility domain to predict the agents' future trajectory and integrate this information in the routing decision. Following the anticipatory mobile networking paradigm [107], an overall increase in terms of robustness and reliability was achieved. PARRoT, as a follow-up protocol, takes the mobility prediction approach and integrates it into a reinforcement learning process that utilizes abstract metrics [481] and represents the routing process [99]. CA-PARRoT extends this with a mechanism to compensate short-term influences and introduces a hybrid machine learning approach where the routing decision still relies on reinforcement learning, but a machine learning component is used to classify a Radio Environment Prototype (REP) and select an appropriate parametrization to achieve the best possible end-to-end performance.

4.4.3.1 B.A.T.Mobile: Leveraging Cross-Layer Knowledge

B.A.T.Mobile introduces a multi-factorial mobility prediction that classifies available information from the mobility control into three levels of assumed accuracy. An iterative prediction process is performed that always chooses the most accurate prediction method for the current step. A prediction width τ is divided into N iteration steps to predict the future position $\tilde{\mathbf{p}}(t + \tau)$. The considered types of information, named in descending order of their accuracy, are:

- The steering vector σ_i describes the current heading to the position of the next iteration step. This information is only available in the very first iteration.
- The currently targeted waypoint $\mathbf{w}(t)$. If the agent's position is in a specific range of this waypoint, it is considered reached, and the next target is used for the remaining prediction process.
- The history of N_e previous positions. Every iteration step appends a new estimate to this list. It is used as a fallback solution to enable a mobility prediction even if no advanced information is available.

The prediction result $\tilde{\mathbf{p}}(t+\tau)$ is then integrated into the periodically broadcasted routing messages of the underlying B.A.T.M.A.N. routing protocol and replaces the former Transmission Quality (TQ) metric for routing decisions.

4.4.3.2 PARRoT: Transition to an Autonomous Routing Process

PARRoT inherits the mobility prediction of B.A.T.Mobile. The routing metric is not solely built on relative mobility; rather, it is gathered by a Q-learning process. Agents share their current and predicted positions, which are then used by the receiving agent to reconstruct their neighbor's trajectory. Further, this is set into a relationship with the agent's own trajectory, and the future availability of a link between these two agents is

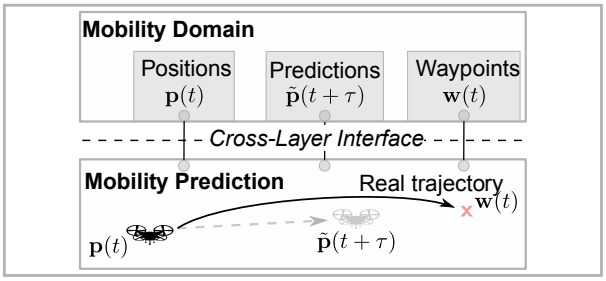


Fig. 4.23: Interconnection of the routing protocol and the mobility domain for multi-factorial mobility predictions. Reprinted from [585].

expressed by the metric $\Phi_{\text{LET}}(d, j)$, representing the Link Expiry Time (LET). Every agent evaluates its own cohesion by comparing its current set of neighbors with a previous set. This produces the metric $\Phi_{\text{Coh}}(j)$, which is shared through the routing messages. The reinforcement learning process is performed by reverse route building. An originator creates a routing message, referred to as chirps in PARRoT, sets the reward value V to 1.0, and broadcasts it. Recipient agents carry out the learning process, considering the originator, which will be the destination d in reverse route building, and the adjacent agent j, from whom the message was received, and process it to

$$Q(d, j) = Q(d, j) + \alpha \left[\gamma_0 \cdot \Phi_{\text{LET}}(d, j) \cdot \Phi_{\text{Coh}}(j) \cdot V_j - Q(d, j)\right] . \tag{4.11}$$

Here, the learning rate α controls the impact of how new routing messages that deliver the reward V_j affect the learned knowledge. The basic discount factor γ_0 guarantees a path degradation. This is of particular interest when all other metrics become 1.0, e.g. in a static network. This measure then forces the shared reward to be decreased and prevents routing loops. The agents maintain a Q-table, where a quality indicator $Q(d, j)$ exists for every destination/gateway pair over which, a chirp has been received. This table is then utilized to feed the known routing table, which enables the packet forwarding process.

Figure 4.24 shows the datagram of a PARRoT routing message that contains identification data, age information, and position information.

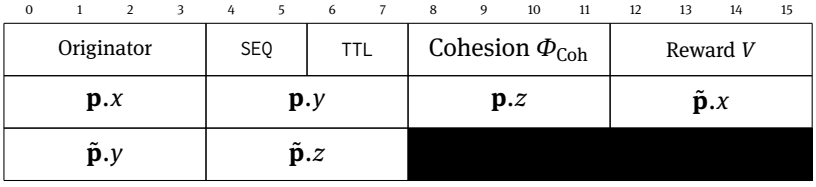


Fig. 4.24: Byte datagram of a PARRoT routing message with a total length of 40 bytes.

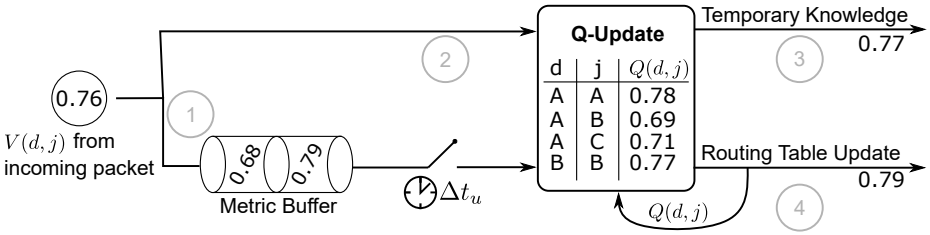


Fig. 4.25: Timer-based update procedure to avoid the short-term effects of having too much impact on learned knowledge. ©[2021] IEEE. Reprinted, with permission, from [586].

4.4.3.3 CA-PARRoT: Introducing a Machine Learning-Enabled Context Adaption

CA-PARRoT proposes further enhancements to PARRoT in order to increase robustness under distinct conditions and reduce the pre-configuration effort. As known from comprehensive simulations, PARRoT has shown a negative impact for time-variant environments as short-term effects harm the accuracy of learned knowledge. B.A.T.Mobile, in turn, proves to be more robust because it buffers incoming information to smoothen the update process. Thus, a timer-based update procedure is also introduced to CA-PARRoT, as shown in Figure 4.25. It is divided into four phases, where an incoming value is pushed to a metric buffer in *phase 1*. The same value leads to an immediate Q-update in *phase 2*, yielding a temporary knowledge in *phase 3* that is used to decide the forwarding of the current routing message. After an elapsed time Δt_u, the best candidate is read from the metric buffer and triggers a Q-update whose result is persisted in the stored knowledge and is also used to update the routing tables in which the packet forwarding is managed (*phase 4*).

Besides the refactored update process, a machine learning component is introduced to obtain a parametrization after the classification of the current environment. For this purpose, different radio environmental prototypes are considered, for which a parameter optimization is carried out in advance, and the best parametrization is provided to the routing protocol. Figure 4.26 shows the adaption approach, which starts with an initial monitoring of the signal strengths of incoming routing messages within an evaluation window. Statistical features are extracted and used to classify a prototype and select its parameters such as the learning rate α, the basic discount factor γ_0, and newly introduced λ and ω, which are used to exponentially weight partial Q-learning metrics.

Random Forests (RFs), Support Vector Machines (SVMs), and Artificial Neural Networks (ANNs) are provided as classification methods. Figure 4.27 shows the classification accuracy that is obtained through a 10-fold cross-validation with the Lightweight Machine Learning for IoT Systems with Resource Limitations (LIMITS) [633] framework, which also enables a model export.

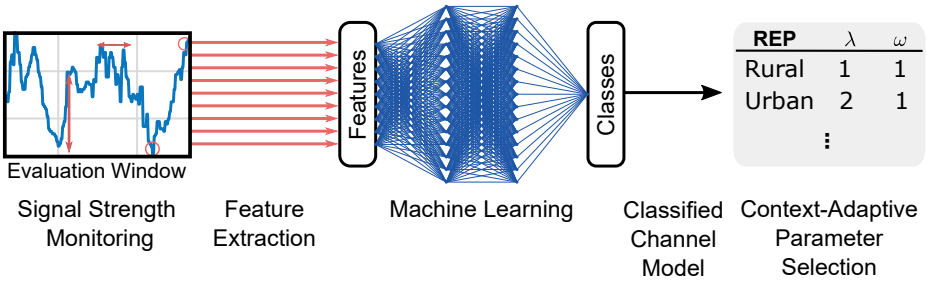


Fig. 4.26: Context-adaption approach for classifiying *radio environment prototypes* (REPs) and selecting parameters accordingly. ©[2021] IEEE. Reprinted, with permission, from [586].

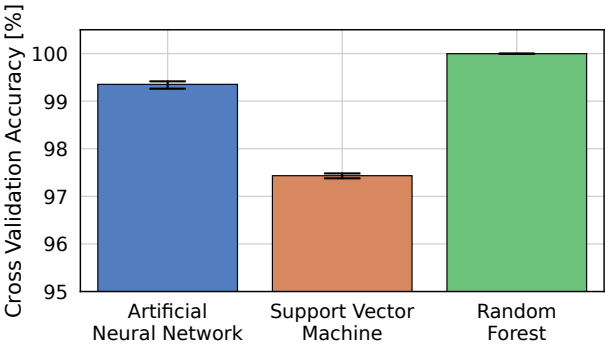


Fig. 4.27: Cross-validation accuracy for the classification of radio environment prototypes. ©[2021] IEEE. Reprinted, with permission, from [586].

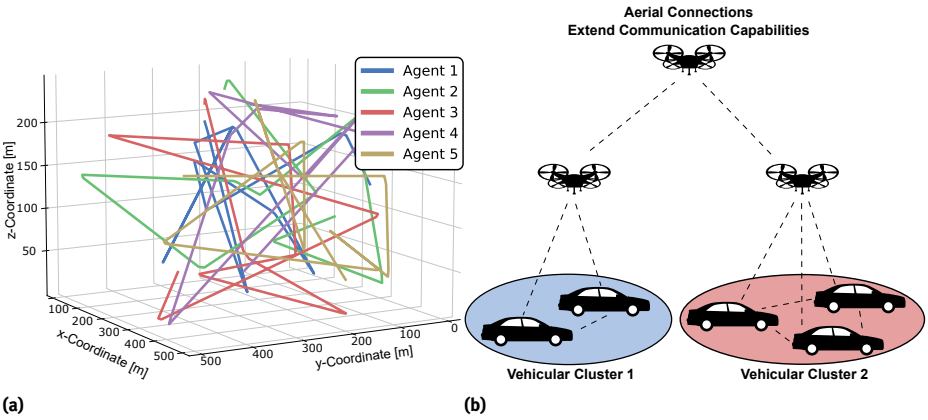


Fig. 4.28: (a) shows the trajectories of 5 agents, following a random waypoint mobility pattern in the three-dimensional reference playground. (b) shows an application-driven mobility, where UAVs fly over clusters of ground vehicles. ©[2021] IEEE. Reprinted, with permission, from [635].

4.4.4 Performance Evaluation of MANET Routing Protocols

In order to carry out a performance evaluation among different routing protocols, a reproducible scenario setup is required that provides a comparable frame for all candidates and makes the overall performance dependent on the evaluated protocol. As Key Performance Indicators (KPIs), the *Packet Delivery Ratio* (PDR) and the mean latency are considered as end-to-end metrics.

A reference scenario is constructed, where ten agents establish a video stream between two agents with a Constant Bit Rate (CBR) of 2 Mbit/s. Besides the communication aspect, the agent mobility has a major impact on the evolution and characteristics of the network topology. Figure 4.28 (a) shows a generic Random WayPoint (RWP) mobility in the three-dimensional playground of the reference scenario. The agents move with a constant speed of 50 km/h and immediately choose their next target when they reach their current one. RWP mobility is considered as a benchmark mobility. Figure 4.28 (b) shows an example of an application-driven mobility, where UAVs hover clusters of ground vehicles to extend their communication capabilities [624]. This mobility pattern involves hybrid types of vehicles that possess inherently different characteristics—a particular challenge for efficient routing protocols.

Another crucial aspect is the choice of a channel model. In rural environments, a Line-Of-Sight (LOS) connection between two agents is usually given. With a lack of objects in the playground, multipath propagation can be neglected, and the free-space path loss L is proportional to the exponentially weighted distance d with $L \propto d^{\eta}$, where η is the attenuation coefficient. This simple but generic model is utilized in the reference scenario for performance evaluation. For other environments, more complex path loss models need to be considered. In urban areas, the impact of objects leads to a higher

importance of multipath propagation, as the LOS path is superposed by reflected signal paths. Therefore, the Nakagami model presents a stochastic impact on the Received Signal Strength (RSS) in addition to the free-space path loss. The empirical analysis of [119] points out commonly used channel models and well-established reference protocols in vehicular research with which the novel protocols are compared.

4.4.5 Results

In this subsection, a comprehensive simulative performance analysis is presented. At first, a scalability analysis is carried out to access the behavior of routing protocols for different types of networks. Afterward, the end-to-end performances are evaluated in scenario studies, that present potential fields of application. To assess an upper bound, an optimal PDR for free-space conditions is introduced, which represents a post-processed analysis of the agents' positions and the theoretical availability of routes. Thus, it is considered a mobility-constrained upper bound.

4.4.5.1 Scalability Analysis

Number of Agents in the Network As seen in Figure 4.29 (a), the number of agents has a significant impact on the PDR of routing protocols. First, for a low number of agents and a low density in the playground, an unneglectable PDR limitation due to mobility can be observed. B.A.T.Mobile, PARRoT, and CA-PARRoT outperform all established protocols for higher density networks. (CA-)PARRoT's course of the PDR is close to the mobility constraint and only shows an impact of routing overhead, which increases for larger scales. Thus, the results show a good scalability, proving (CA-)PARRoT to be suitable for high-density networks.

Impact of Speed on the End-to-End Performance MANETs are characterized by their mobility, which can mean high agent speeds in many cases. Figure 4.29 (b) shows the performance for a range of slow-moving agents up to highly mobile scenarios. High speeds require routing protocols to adapt to the network topology very quickly. For speeds over 100 km/h, most routing protocols fail to provide reliable routes, which causes the PDR to drop and appear undisclosed. Only (CA-)PARRoT, which is affected by the higher requirements of increasing speeds, is capable of providing high PDRs by anticipating the network topology and compensating link losses.

4.4.5.2 Scenario Studies

UAV Communication in a Rural 3D-Playground As seen in Figure 4.30, the novel routing protocols, B.A.T.Mobile, and (CA-)PARRoT, outperform the established reference protocols. The study considers a rural three-dimensional playground with an air-to-air communication between two UAVs. As the communication range does not cover the

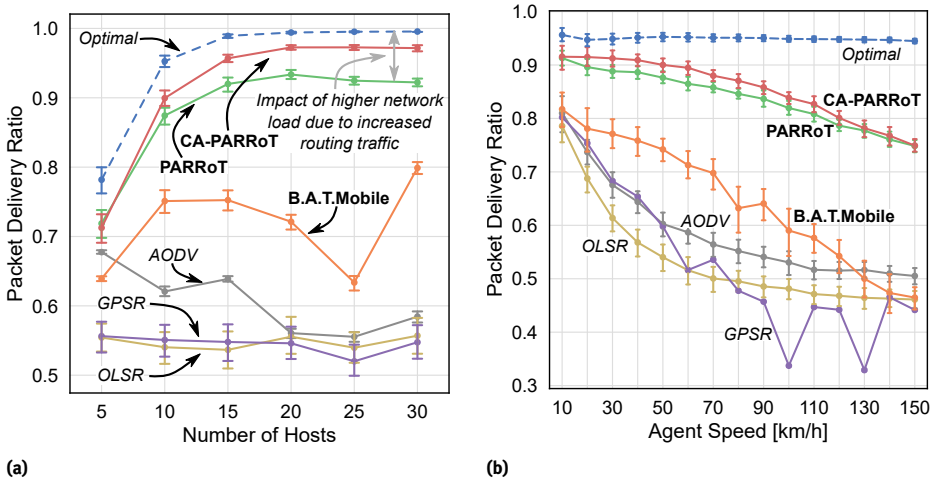


Fig. 4.29: Scalability analysis of routing protocols. PARRoT and CA-PARRoT outperform the considered reference protocols significantly and, thus, provide a robust communication behavior even for challenging conditions.

whole playground, a mobility constrained optimum is calculated, as a routing path is not necessarily available for every point of time.

All established protocols fail to provide a reliable PDR above the 70 % mark, which allows CA-PARRoT to outperform them by 48 % in means. CA-PARRoT achieves a 3 % and 19 % higher PDR than PARRoT and B.A.T.Mobile, respectively, and is thus the best-performing mobility-predictive protocol, with only a 5 % gap to the theoretical upper bound. The proposed protocols, therefore, show gradual improvements with every development stage. Also, considering the latency, CA-PARRoT performs best and shows a 21 % reduced latency compared with OLSR, which is the lowest latency established protocol in this analysis.

Challenging Conditions in Urban Areas Figure 4.31 shows the end-to-end performance for urban radio conditions. B.A.T.Mobile and PARRoT outperform the reference protocols for both KPIs under consideration. PARRoT's performance is a bit weaker than that of B.A.T.Mobile due to the immediate impact of incoming routing packets, where each packet is used for updates, and short-term effects compromise the learning accuracy. The proposed CA-PARRoT is able to avoid this behavior and achieves the highest reliability of all considered routing protocols. Also, in terms of latency, CA-PARRoT shows up to 51 %, lower values compared with established protocols and a 9 %, lower latency compared with that of the B.A.T.Mobile. In general, higher latencies can be observed. These are caused by spontaneous link losses that enforce more buffering in the MAC layer.

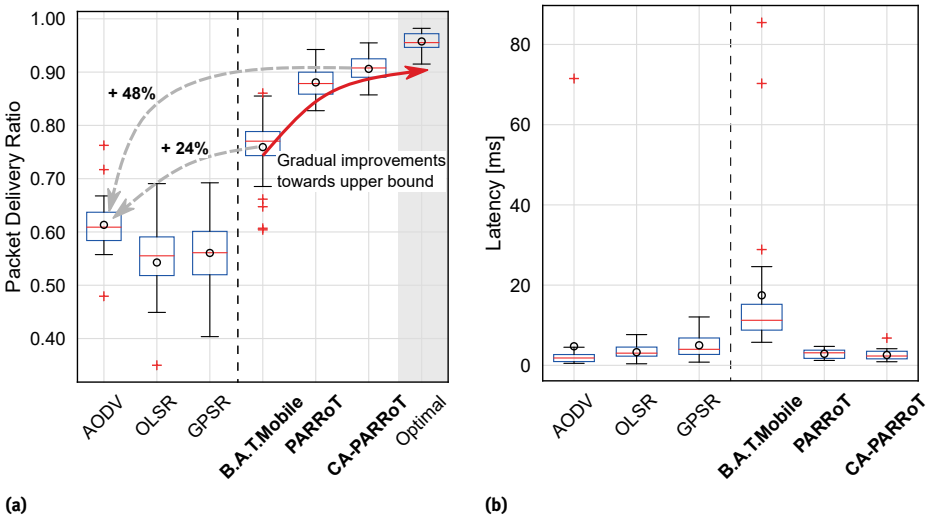


Fig. 4.30: The mobility-predictive routing protocols outperform the established references in terms of reliability, expressed by the PDR. The proactive integration of trajectory knowledge establishes a robust route finding and advances the performance towards the mobility-constrained optimum. ©[2021] IEEE. Reprinted, with permission, from [586].

Performance of MANET Routing in Application-Driven Scenarios The previously presented analysis respects different scalability and radio propagation influences but uses a generic random waypoint mobility model. As real-world applications of MANETs may have mobility in coincidence with the corresponding task, two examples are studied in the following.

Aerial Cluster Hovering

UAVs are used to hover over clusters of cars. In this analysis, ten UAVs are deployed to cover a total of 50 cars, of which ten are equipped with communication interfaces. The remaining 40, therefore, impact only the cluster selection and mobility of other cars, but not the communication system. Figure 4.32 (a) shows the PDR of this scenario. The incremental position updates reduce the accuracy of the mobility prediction and cause B.A.T.Mobile to have high performance losses. However, PARRoT also uses mobility prediction, but is still able outperform all other protocols due to a considered cohesion-aware metric in the learning algorithm.

Distributed Dispersion Detection

To explore plumes, random mobilities are not effective. The distributed dispersion detection (DDD) is a mobility model that is aware of maintaining the cohesion of the UAV swarm during exploration and, therefore, provides a high amount of available

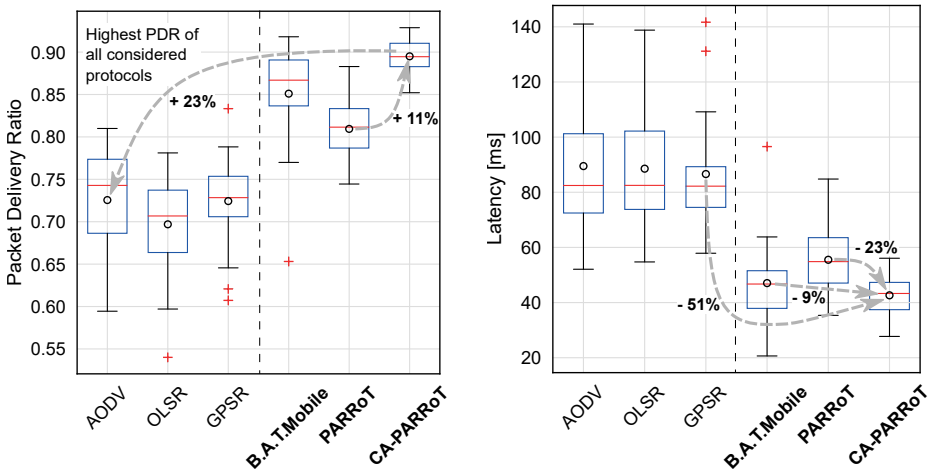


Fig. 4.31: End-to-End performances in a three dimensional playground with urban radio conditions. ©[2021] IEEE. Reprinted, with permission, from [586].

routes. This is also reflected in Figure 4.32 (b), as all routing protocols show a similar, high performance. Nevertheless, B.A.T.Mobile and PARRoT are able to reduce negative outliers and provide a more reliable communication.

4.4.6 Conclusion

The results show that the proactive integration of mobility-domain knowledge enables a significantly more robust behavior, which can outperform established routing approaches in a vast variation of challenging conditions. The utilization of machine learning and reinforcement learning adds an additional gain and robustness, as the comparison between B.A.T.Mobile and (CA-)PARRoT has shown. The more the routing protocols are aware of their environment, e.g., its mobility and radio conditions, the higher the achievable robustness becomes. Therefore, high KPIs could be observed, even in high-scale scenarios. Intelligent routing algorithms are a key component for the realization of efficient infrastructure-less device-to-device communication, not only for WiFi-based networks but also for other technologies such as cellular approaches. The lack of a centralized unit promises lower latency and enables the network participants to learn and adapt to their situation based on local observations, reducing the need for communication expensive status updates.

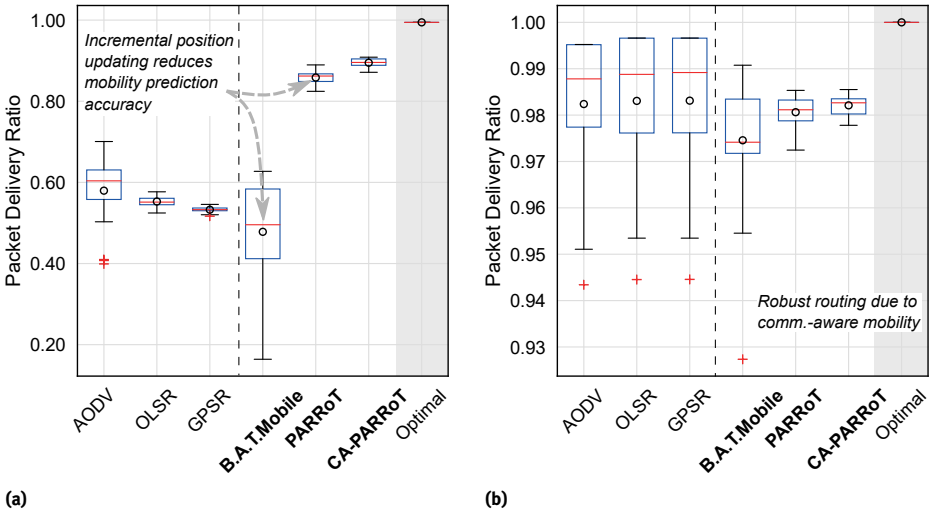


Fig. 4.32: Achieved packet delivery ratios for application-driven scenarios.

4.4.7 Acknowledgments

In addition to the CRC 876, part of this work has been supported by the Ministry of Economic Affairs, Innovation, Digitalization and Energy of the state of North Rhine–Westphalia in the course of the Competence Center 5G.NRW under grant number 005–01903–0047, and in the course of the project Plan Play under grant number 005-2008-0047.

4.5 Modelling of Hybrid Vehicular Traffic with Extended Cellular Automata

Michael Schreckenberg
Tim Vranken

Abstract: Vehicular traffic is a complex system with multiple challenges. For example, highways and urban traffic networks, different vehicle types with varying maximum velocities (cars, trucks, public transportation, etc.), and varying driving behaviors each impacts traffic flow uniquely and strongly. This means that in order to minimize the number of congestions and the average travel times, it is necessary to analyze, model, and simulate traffic in multiple different scenarios.

In the following, we will introduce cellular automaton models for different scenarios. These cellular automaton models aim to reproduce macroscopic traffic phenomena through microscopic simulations. With the help of these simulations, we are able to analyze, understand, and predict traffic in the given scenarios. Lastly, based on the predictions, we can attempt to simulate the same scenarios with small adjustments in order to maximize traffic flow and minimize travel time.

To this end, we will start with introducing and analyzing highway traffic. Here we will focus on applying real-life weather data and how it impacts traffic flow. Next, we will investigate, where the limited space and regular interruption of urban traffic flow by traffic lights and intersections result in new and additional constraints. Lastly, communicating and automated vehicles will be introduced into the simulations. The different reaction times, behaviors, and the human-robot interaction are expected to result in new challenges that have to be investigated and predicted.

4.5.1 Introduction

The topic of vehicular traffic is gaining more and more attention amid rapidly growing numbers of vehicles on the road and increasing amounts of traffic congestion, leading to longer average travel times and more fuel consumption. The road capacity cannot be increased indefinitely through the addition of new lanes, which means that other methods to increase road capacity or use it more efficiently need studying.

In order to use the road more efficiently, traffic and congestion have to be understood better. To this end, we will first analyze and model traffic behavior on highways in Section 4.5.2 with two main goals. The first is to understand the creation of jams and their influence on the traffic flow more deeply. For this, we will use the model by Nagel and Schreckenberg [467]. The second is to model the asymmetric lane changing

required in multiple countries around the world. For this, we will use the one-lane model by Lee et al. [379]. Asymmetric lane changing means that agents are supposed to drive on the most outer (right) lane unless they overtake a slower vehicle.

After analyzing some general traffic flow behavior, we take a look at two more specific problems, the influence of weather on highway capacity, and the missing data of traffic detectors for real-time applications.

While these analyses allow better traffic flow predictions on highways, the situation in urban and inner-city networks is more complicated due to a higher degree of interactions of crossing flows and a regular interruption of the traffic caused by traffic lights. Therefore, in Section 4.5.3 we adapt the model by Lee et al. to include driving behavior before traffic lights and in intersections. Furthermore, traffic flow, based on empirical data from Düsseldorf's inner city, is analyzed. There, different analytical real-time routing methods are applied to minimize traffic jams and the average travel time of the agents. These analyses help identify traffic bottlenecks that have a high impact on the creation of traffic jams and the increase of the average travel time. However, space in urban and inner-city networks is often very restricted, so it is not possible to increase the capacity of these bottlenecks by building more lanes. Due to this, in a follow-up simulation, we changed one of these bottlenecks in a way that two lanes (one leading in and the other leading out of the city) at the chosen intersection were dynamically changed to either lead into or out of the city. This way, the road capacity could be dynamically adjusted to fit the changing demand of commuters.

Lastly, in Section 4.5.4, we simulate heterogeneous traffic with automated and human-driven vehicles. It is expected that automated vehicles will reduce or even eliminate traffic jams at 100 % penetration. However, it will take multiple decades until 100 % is achieved, and the impact of automated vehicles in heterogeneous traffic is unclear due to the different behaviors of automated and human-driven vehicles. Due to this, we adapt our model to simulate the different behaviors of automated, communicating automated, and human-driven vehicles. The goal is to simulate heterogeneous traffic where the three different vehicle types mix and then predict how this will impact traffic flow and road capacities.

4.5.2 Highway Traffic Data Aggregation

The analysis of highway traffic flow can be divided into two topics. General traffic behavior and the real-time traffic situation will be analyzed in this section. For that, we will first model and simulate how traffic jams are created to understand better how and when free-flowing traffic transitions out of free flow due to traffic jams. After that, empirical data is used to create realistic lane-changing behavior while considering asymmetric lane changing rules, as they are applied in countries like Germany or France.

After that, the impact of real-time weather data on jam creation and travel velocities will be analyzed, and the results will be used to increase traffic-flow predictions. The prediction of real-time traffic requires real-time data, which sometimes can be missing due to failing detectors or communication channels. Therefore, the last part of this section will deal with the problem of how to replace missing data accurately.

4.5.2.1 Traffic Jam Analyses

The goal of traffic research is often to prevent traffic jams, or reduce their lifetimes. To this end, Bette et al. [59] analyzed traffic jams using the Nagel-Schreckenberg model [467]. For that, the traffic density was determined at which free-flowing traffic transitions into jammed traffic based on a stability criterion. Afterward, the ratio of jammed cars was separated into different mechanisms, the jamming rate, jam lifetime, and jam size. It was shown that small jams already occur at very low densities and that the increasing life-time of these jams at higher densities is what leads to the transition of the traffic flow from free-flowing to jammed traffic. Furthermore, exponents that control the scaling of all three jam mechanisms close to the critical density have been derived from random walk arguments.

4.5.2.2 Asymmetric Lane-Changing Rules

Lane changing in many countries is asymmetric because drivers are required by law to drive on the most outer (right) lane as long as they do not overtake a slower vehicle. This asymmetric driving behavior creates multiple differences compared with highway traffic without overtaking restrictions. For that reason, empirical data from two countries with such asymmetric rules (Germany and France) have been considered in [248]. A multi-lane cellular automaton model with asymmetric lane changing rules has been created and calibrated based on this empirical data.

This model is based on the one-lane model by Lee et al. [379], where agents have different driving behaviors and a maximal deceleration capability. These two points together allow the model to reproduce accidents due to miss behavior. Because of this, the lane-changing rules have to fulfill three functionalities. Firstly, they have to be safe, which means that the distance to both the preceding and to the following agents has to be large enough, depending on the current velocity. Secondly, the agents should change to an outer lane (one on their right) as soon as the lane change is safe and they do not have to decelerate, while they change to an inner lane (one on their left) only if it is safe and they can accelerate or prevent a forced deceleration. Lastly, the agents have to be prevented from overtaking a slower driving vehicle on a more outer lane.

After adding rules to ensure all three points, the model is able to reproduce empirical lane usage for two- and three-lane highways. Furthermore, a higher number of lanes can be simulated easily after one parameter of the model is re-calibrated accordingly.

Tab. 4.5: Maximum vehicular traffic flows obtained from the whole empirical dataset. Here, w represents the mean of all available water film sensors at a certain point of time.

Surface	Range of water film thickness w	Maximum flow
Dry	$w \leq 0.15$ mm	1960 vehs/(h, lane)
Damp	$0.15 \text{ mm} < w < 0.9$ mm	1720 vehs/(h, lane)
Wet	$w \geq 0.9$ mm	1320 vehs/(h, lane)

4.5.2.3 Influence of Weather Data on Traffic Predictions

Vehicle-2-X communication has grown rapidly within the past two decades, which has increased the availability of extended Floating Car Data (xFCD) that can be applied in the field of traffic information and improvement. One possible integration of this additional real-time data is the inclusion of weather data in traffic-flow predictions [246]. In order to identify the current weather on the road, the water film thickness is taken by local weather stations. Vehicles equipped with xFCD are able to gather and communicate this data through the use of rain-sensing windscreen wipers, which react to water spray. While this vastly increases the available data, floating car data can be more unreliably due to the limited radio spectrum available to transmit this data. An efficient way of communication is developed in Section 5.2 to transmit data reliably and efficiently. An analyses of the correlation between the average velocities of passenger cars and water film thickness on the road showed a strong negative correlation of up to -0.4 for rush-hour traffic. This means that the incorporation of weather data into traffic information systems is expected to be exceptionally beneficial for commuters. Furthermore, Table 4.5 shows that an increased water film thickness also decreases the minimal road capacity above which the traffic flow becomes unstable and can transition away from free-flow, which indicates that the inclusion of weather data improves traffic-jam predictions.

After the impact of water film thickness on the traffic flow was analyzed, the new insights were added to the previously discussed asymmetric multi-lane version of the Lee et al. model. For that, an additional dallying parameter $p(w)$, which depends on the water film thickness on the road w, was introduced. A higher value of this parameter increases the probability of an agent decelerating even if the leading vehicle is far enough away for its velocity to be safe. The results in Figure 4.33 show a good agreement with the empirical data for the roughly 6 km-long Autobahn section chosen for the study. This also shows that an accurate traffic prediction needs reliable real-time data in order to work. However, empirical data is often not reliable enough to ensure constant real-time updates, and data can be missing. This missing data then has to be replaced by approximations in real time.

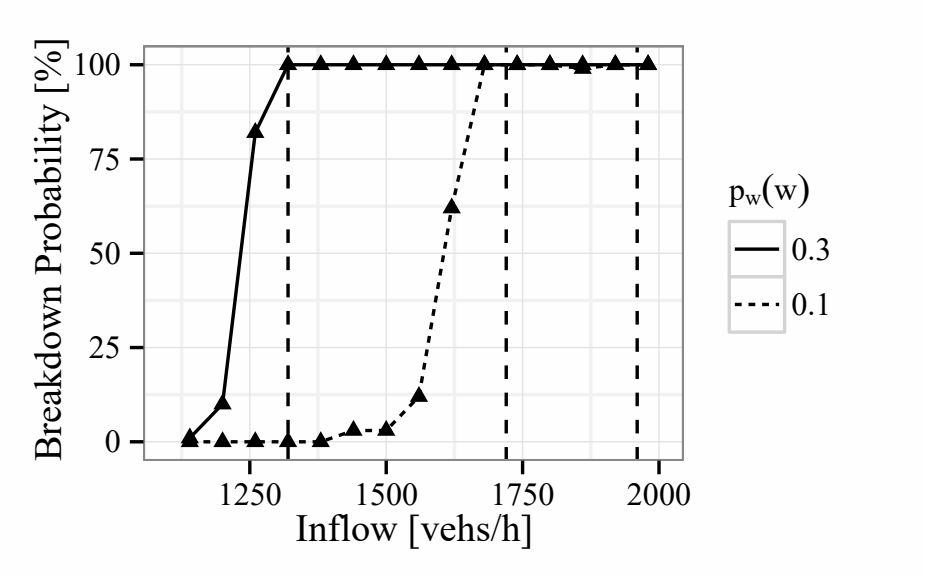


Fig. 4.33: Simulated breakdown probability for two different degrees of surface wetness.

4.5.2.4 Replacing Missing Data

Previously, in the exponential smoothing prediction method [136], missing data of a traffic detector was replaced analyzing the historical data of this detector. This has two major downsides. Firstly, the historical data ideally has to be from the same day and time in the past 30 weeks of all D detectors. If, for example, a detector stops communicating its findings on a Monday at 12 o'clock, then the data of 12 o'clock of the past 30 Mondays has to be considered. This means that all data points of all D detectors have to be saved for at least 30 weeks in order to make accurate predictions on missing data points. The second problem is that the traffic situation ten weeks before, for example, does not have to be the same as today. Accidents or road works, for example, could shift the traffic flow from one street to another, which would strongly increase the error of such predictions. Because of this, a new method to replace missing data has been introduced in [247].

In this new method, 60 minutes of historical data of the surrounding N detectors from the preceding week is taken. This data is then used to train a Poisson Dependency Network (PDN) [249] (which is a form of a Poisson model explained in Section 4.1.2.6). This PDN shows how strongly the traffic-flow data at one detector point correlates with the data of the detector that has missing data. Then, in a final step, the real-time data of the other d detectors is inserted into the PDN to fill the missing data point.

In order to test the method, empirical data was taken, and a prediction of it was made as if it were missing. Then the empirical data was compared with its prediction in Figure 4.34. One can see that the PDN is closer to the actual data than the exponential

smoothing prediction that just used the past N weeks of traffic data of the "missing" detector. A general test at different times of the day and different days of the week showed that the PDN not only uses less historical data but also predicts missing traffic data more accurately. A more in-depth analysis of the problem of traffic flow prediction is given in Section 4.1.

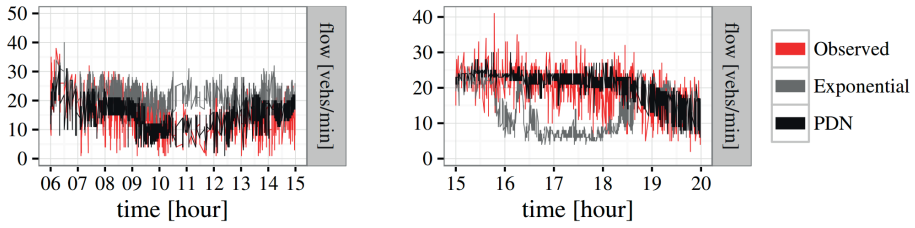


Fig. 4.34: Example empirical data together with exponential smoothing and PDN predictions based on the data.

4.5.3 Urban Traffic Simulations

Inner-city traffic is more complicated than highway traffic due to more interactions of vehicles with different travel directions and intersections with or without traffic lights. In order to analyze and simulate inner-city traffic, the Lee et al. model, which was used as a basis for highway traffic simulations in the previous section, was modified with additional rules in [710] to reproduce empirical intersection traffic data. In another work, different methods to dynamically optimize inner-city traffic through different routing methods and a dynamical application of lanes were analyzed [709]. There, we were able to show that while traffic flow is often above the network capacity, one can decrease the number of traffic jams and the average travel time through more dynamic routing methods and the use of the infrastructure.

4.5.3.1 Cellular Automaton Model

The simulation of urban traffic with a model based on that by Lee et al. [379] has to include a couple of complex situations. Different intersections, for example, can be very different in their structure and serve different purposes. Some intersections can have lanes on which one is only allowed to turn left, while another intersection with the same number of lanes and roads allows turning left or keeping going straight from the most inner (left) lane. Furthermore, agents have to ensure that they arrive on the right lane, depending on their route before they arrive at the traffic lights. Finally, vehicles that turn left within an intersection sometimes have to take into account that the crossing

traffic has a green light at the same time, and they are only allowed to turn if they do not disturb this ongoing traffic flow.

After these points were included within a cellular automaton model in [710], the model was calibrated so that the time requirement $t_B \approx 1.6 \pm 0.6$ s is in accordance with empirical data. To calculate the time requirement t_B, the number of cars that arrive at a red traffic light in front of an intersection within the simulation is set higher than the number of vehicles that can pass this traffic light once it turns green. When the traffic light turns green, the time each agent requires to move into the intersection after the preceding agent moved into it is taken. This time is the time requirement t_B and, after 3 – 4 s of a green traffic light, it averages out to around $t_B = 2.23 \pm 0.04$ s. This is in accordance with the empirical data.

Afterward, an intersection was modeled in that the traffic lights that control left-turning traffic have green at the same time as the crossing traffic flow. Because turning agents decelerate before they turn, the time requirement is already higher than for traffic that does not turn, which means that the lane capacity is lower. Furthermore, this time requirement increases strongly in correlation with the crossing traffic flow. This means that in order to minimize traffic jams and travel times in urban traffic, one not only has to consider the traffic flow from one point to another but also all other traffic flows within the network.

4.5.3.2 Inner-City Traffic Optimization

In [709], inner-city traffic was analyzed and simulated with the help of empirical data from Düsseldorf's inner-city. Based on the analyses of the traffic flow, different routing methods were used to guide the traffic through the application of real-time data. The analyses showed that the traffic capacity of roads inside the city is not enough to cover demand at all times. However, a routing method that aims to make maximal use of the road capacity rather than route vehicles depending on their travel time would improve the traffic flow significantly and could reduce the average travel time by up to 23 %. The downside of optimizing the traffic depending on the network capacity rather than the travel times is that if the traffic flow is below the road capacity, vehicles will take roads with longer travel times than necessary, which increases fuel consumption and travel times unnecessarily. Due to these findings, a new routing method was developed that considers both the road capacity and the travel time of each agent individually. As one can see in Figure 4.35, the new routing method (green) recreates shorter travel times than the network optimization method (red) at low traffic volumes while also reducing the travel times over routing methods where each agent uses the route with the shortest travel time (blue and purple) at high travel volumes.

After it was shown that a more efficient routing method would decrease the average travel time and make more use of the given road capacity, we tested how the road capacity itself could be dynamically optimized. To this end, one of the busiest intersections leading into and out of the inner city was identified. This intersection

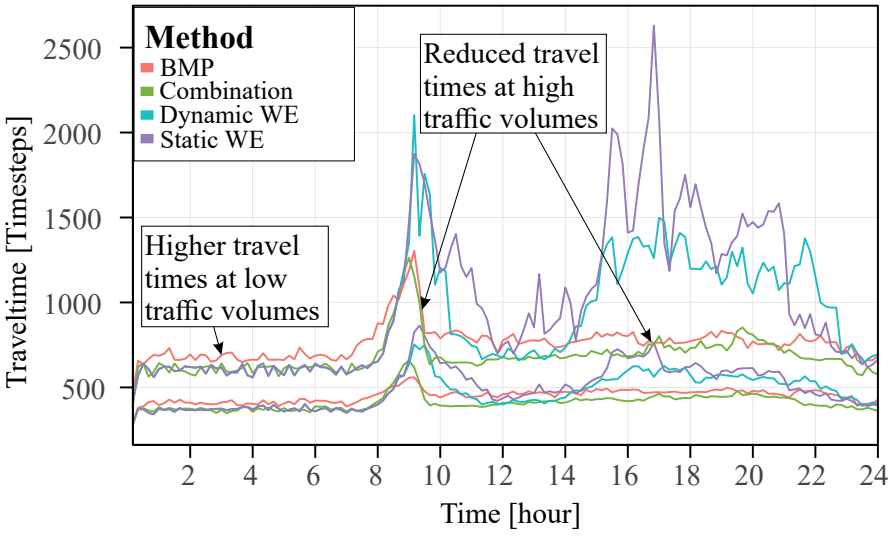


Fig. 4.35: Travel times for different traffic assignment methods. The lower lines represent the average travel time taken over 10 minutes while the upper line represents the 95 % confident interval upper bound.

connects the city ring with one of the main roads leading through the city. One of the lanes on the city ring, as well as one of these on the road leading through the city, were changed to dynamic lanes. Traffic into and out of Düsseldorf is very unbalanced at different times of the day due to the high number of commuters. The dynamic lanes were able to reduce the commute time into the city in the morning while also decreasing the travel time needed to leave the city in the afternoon by increasing the road capacity where it was needed and decreasing it where it was not needed. Through this dynamic change done to a single intersection, the average travel time could be reduced by over 10 % without changing the way vehicles currently choose their routes. Note that these travel-time reductions aren't necessarily the optimal reductions. The goal is rather to understand the network and find its bottlenecks. How to find the optimal routes for each vehicle to reduce the global average travel time is analyzed in Section 4.1.4. There, a new method that applies a reinforcement learning algorithm is simulated, and the results are compared against others.

4.5.4 Automated Vehicular Traffic Flow

Even though dynamic routing and dynamic shifting of the road capacity can reduce the average travel time, this is not a permanent fix for the increasing number of traffic jams

since the number of vehicles is expected to continue increasing in the near future. By contrast, automated vehicles in an 100 % automated traffic are expected to increase road capacity, reduce travel times, fuel consumption, and accidents significantly. However, it is not clear yet how automated vehicles will be introduced into the traffic. Their impact on the traffic dynamics in heterogeneous traffic, when they mix with human-driven vehicles, has been widely discussed. Traffic will be heterogeneous for multiple decades due to an average lifespan of a vehicle of around ten years [42].

In order to predict heterogeneous traffic behavior, a new cellular automaton model was introduced in [708]. One of the big challenges of cellular automaton automated vehicle traffic is that Automated Vehicles (AVs) and Communicating Automated Vehicles (CAVs) have a reduced reaction time compared with human-driven vehicles (HVs). For CAVs, this reaction time could go as low as the time it takes to communicate, which is currently around 0.1 s but could go even lower with 5G, which is expected to become an important method for future connected and automated vehicles. Currently, communication does not always take a fixed time length but instead varies depending on the limited available radio spectrum. If the full spectrum is used, communication can take a lot longer than 0.1 s, depending on the means of communication. An in-depth analysis of the problems of communication and how it is realized is given in Section 5.2. For the remainder of this section, we will assume a stable communication with a 0.1 s communication time. Therefore, the first step toward simulating AVs or CAVs with cellular automaton models was to reduce the time-step length of the model to 0.1 s per time step.

4.5.4.1 Reduced Time-Step Length

The cellular automaton model introduced in [708] is based upon the Pottmeier et al. [508] accident-free version of the Lee model [379]. Human-driven vehicle agents in this model can judge their situation optimistically or pessimistically. If they judge it optimistically, they do not expect their leading vehicle to decelerate strongly (only dawdle). In this case, they can follow the leading vehicle with less than the minimum safety distance, something that is often found in empirical data. If they judge their situation pessimistically, they expect the leading vehicle to decelerate at any moment. They do not follow with the minimum safety distance but even apply an additional safety distance, depending on their velocity.

The newly introduced model reproduces this behavior while also having a 0.1 s long time step and keeping an average reaction time of 1 s for the HVs. For that, multiple changes were made to the calculation of the safe velocity, dawdling, and the judgment of the situation. However, Figure 4.36 shows that the resulting 0.1 s long time-step model presented in [708] is still able to reproduce realistic human-driven vehicle traffic comparable with the model presented by Lee et al. [379] and the modified version of that by Pottmeier et al., which are both known to reproduce empirical traffic well [508]. As one can see, the main difference between the three models is at a density of $20 - 45\ \frac{\text{Veh}}{\text{km}}$

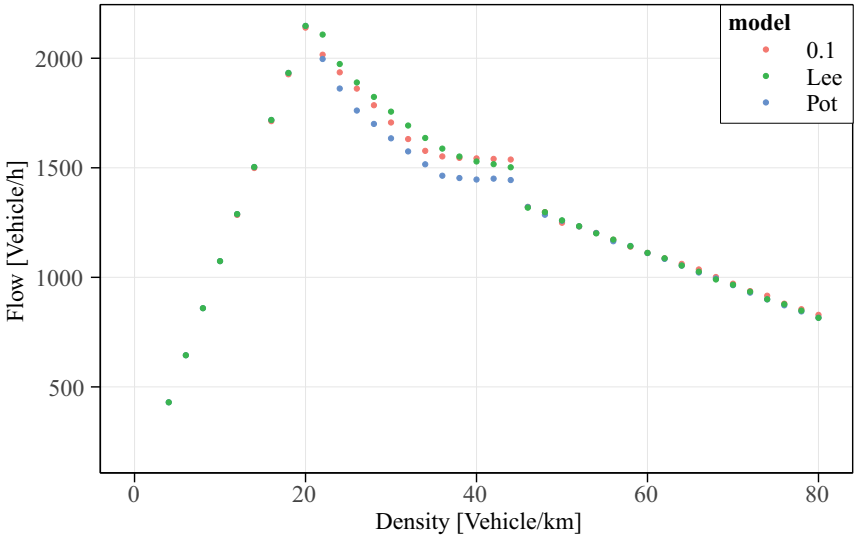


Fig. 4.36: Fundamental diagram for the three different human-driven vehicle cellular automaton model.

in the so-called synchronized traffic phase [317]. Pottmeier et al.'s modification of the Lee model only differs from the Lee model in the calculation of the judgment of the situation. Agents in this model are less likely to be optimistic, which prevents accidents but also increases the average vehicle following time and so reduces the traffic flow. The reduced time-step length also uses this curbed optimism due to which the traffic flow is initially below that of the Lee model. However, due to changes to the dawdling, the velocity distribution is more uniform in the 0.1 s time-step length model, which strengthens the synchronization and increases the traffic flow towards the end of this traffic phase.

Overall the differences between the three models are smaller than the fluctuations observed in empirical traffic, and they all reproduce realistic traffic flow.

4.5.4.2 Heterogeneous Automated Vehicle Traffic

AV and CAV agents have multiple differences compared with HV agents. Their three most important differences are reaction time, dawdle, and behavior calculation. An AV has a reaction time of 0.5 s and a (CAV) one of at least 0.1 s compared with the 1 s of HVs. Neither CAVs nor AVs dawdle at all, while HVs dawdle with a probability of up to 37 % [379]. Lastly, CAVs and AVs do not judge their situation optimistically or pessimistically, but instead, they always follow the leading vehicle with at least the minimum safety distance. If the leading vehicle is human-driven, then they follow with more than the

minimum safety distance to be able to react to unpredictable human behavior without creating strong deceleration waves.

These differences between AVs, CAVs, and HVs were defined and included in the model in [708] before heterogeneous traffic flow was simulated. The simulation results are shown in Figure 4.37, together with the theoretically predicated capacity increase [215] for heterogeneous traffic. One can see that automated vehicular traffic is expected to increase the traffic capacity compared with homogeneous human-driven. The effect is even stronger for communicating automated vehicles. A reduced reaction time means that the minimum safety distance (and so the average vehicle following time) is lower, which allows higher traffic flow at similar densities. Furthermore, the reduced reaction time also means that these automated agents overreact less to human dawdling, which reduces deceleration waves. This effect is strengthened because they do not apply optimistic or pessimistic behavior states. If an HV agent changes its behavior from optimistic to pessimistic, it needs a higher distance to its leading vehicle even if the velocities would not change. This means that the agent has to decelerate more than AVs or CAVs that only have to decelerate as much as the leading vehicle to keep up the same distance.

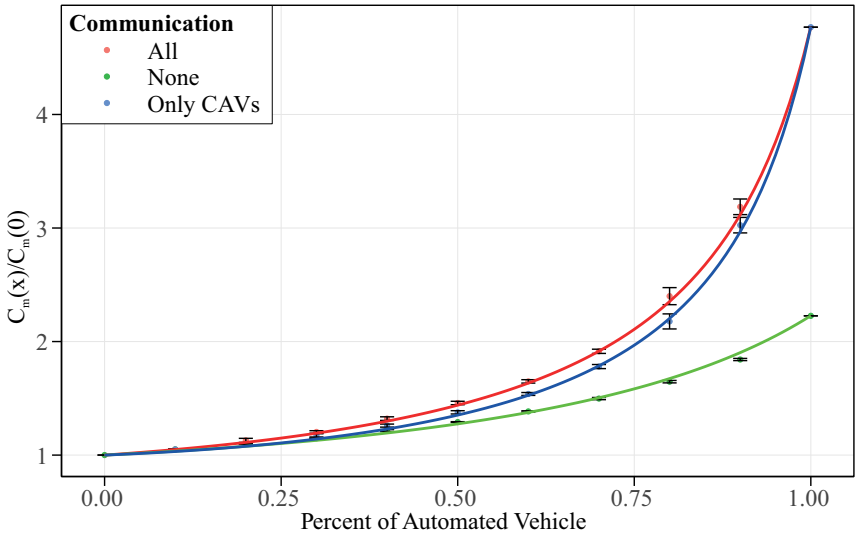


Fig. 4.37: Road capacity increase over homogeneous human-driven vehicular traffic depends on the percentage of automated or communicating automated vehicles.

However, while these results show an improved road capacity in every heterogeneous traffic situation compared with purely human-driven traffic, the model was also able

to reproduce an increased rear-end-collision risk due to human-robot interactions. Human drivers tend to drive with less than the minimum safety distance if they judge the traffic situation optimistically and do not expect to have to decelerate within the next couple of seconds. Before, when they followed another HV agent or even an AV agent, the remaining safety distance the agent used was just enough to prevent accidents if the traffic situation suddenly changed. Now, however, because the CAVs are able to react with only one-tenth of the time the average human needs, the velocity difference between a decelerating CAV and the following HV is so much larger than before that the following HV is not able to prevent an accident once the agent has reacted.

This shows that the different behaviors of HVs and CAVs have the potential to increase the rear accident risk. However, the results shown in Figure 4.37 are simulated after this accident risk was prevented through a change in the behavior of CAVs. If an HV agent n follows a CAV agent $n + 1$ with less than the minimum safety distance, then the CAV agent increases its distance to its leading agent $n + 2$. This way, if agent $n + 2$ decelerates, then agent $n + 1$ can decelerate after driving this additional distance, which gives agent n the time to react to the brake lights of agent $n + 2$, thus preventing accidents.

4.5.4.3 Conclusion

In this section, we highlighted different analyses of vehicle traffic scenarios on highways, in urban areas, and among heterogeneous traffic where automated and conventional vehicles are mixed. The main goal of those analyses is to understand and predict traffic better. Furthermore, the already existing cellular automaton vehicle model introduced by Lee et al. [379] was modified for multiple occasions.

We were able to show that traffic jams already form at very low densities and that the increase in their duration leads to a transition of free-flowing to jammed traffic. While a critical density from that on free flow could transition to jammed traffic flow, it was also shown in other works that this critical density is not fixed and can vary. On highways, the local weather in the form of rain (measured by water film thickness) was identified as an important influence on traffic flow, while urban traffic is mostly dominated by traffic lights. Both problems were analyzed through the use of real-life data as well as simulations of a modified Lee model.

Through the results of those analyses, the accuracy of traffic forecasting could be improved as long as enough real-time data was given. Unfortunately, the reliability of empirical detectors is often not fully guaranteed, and detectors can malfunction or not communicate data. For such cases, we have introduced a new method to fill missing data gaps that uses less historical data and is more accurate. Lastly, we modified the Lee model to predict the impact automated and communicating automated vehicles will have in future traffic when mixing with conventional vehicles.

4.6 Embedded Crowdsensing for Pavement Monitoring and its Incentive Mechanisms

Maximillian Machado
Ran Ran
Liang Cheng

Abstract: Due to its pervasiveness and convenience, crowdsensing is regarded as an effective method to collect specific data. This section surveys projects that take advantage of embedded crowdsensing to collect pavement condition data and describe how crowdsensing platforms conduct road damage detection using deep neural networks with images captured with smartphones. Before such discussion, we explore how to motivate users to participate in low platform-cost crowdsensing tasks. Our research models the pavement crowdsensing problem and designs new incentive mechanisms based on a platform-driven greedy algorithm. Through extensive simulations, the performance of the incentive mechanisms is evaluated and compared in different scenarios in terms of the platform cost and the overall task completion time. The best of them can reduce the total completion time by half compared with the reverse auction incentive mechanism. We conclude this contribution with future work discussions.

4.6.1 Introduction

As road infrastructure increases in size and complexity, innovative solutions must be developed to cope with road degradation. With the current methods used to collect road condition information, covering 4.18 million miles of road in the U.S [555] is a difficult undertaking. One viable solution is to create a crowdsensing platform where smartphone users collect road pavement data. In this case, users could automatically detect and report poor road conditions using embedded cameras, accelerometers, and 4G/5G networks. However, such a network would require an active user base and a means of maintaining crowdsensing participation. Thus, crowdsensing schemes must design an incentive structure for continuous user activity. A crowdsensing platform utilizes an incentive mechanism to motivate user participation, produce diverse data pools, and generate quality information.

Later in this section, we will formally express the incentive mechanisms as one of the key components of crowdsensing platforms. The mechanisms designed in this section are evaluated through experimental procedure, and vary based on the reward distribution assigned to sensing tasks. Favorable mechanisms are those that exhibit both low *platform cost* and *total operation time* when completing sensing tasks. Note, these criteria are not the only metrics for measuring incentive mechanism efficacy.

However, *platform cost* and *total operation time* metrics are sufficient for providing a solution framework to the purposed crowdsensing problem.

Our crowdsensing platform is parameterized with nine unique incentive mechanisms, and aims to collect pavement condition data for varying percentages of a fixed area with economic feasibility in mind. All mechanisms are based on a platform-driven greedy algorithm that motivates users to select sensing tasks that can provide the highest net profit margin for the participant. Eight of the nine incentive mechanisms are uniquely defined in this section while an additional incentive mechanism is motivated from recent related work in crowdsensing literature. Our results provide guidance in selecting the best incentive mechanism in different settings of pavement crowdsensing.

Here are the key contributions of the section:

- The incentive mechanisms we design can effectively avoid the cost explosion problem as users choose their sensing tasks before starting to work on them. Thus, sensing tasks can only be committed to by one participant at a time. Cases where a sensing task should be reexamined–possibility due to poor readings–can be addressed by modeling repeated sensing tasks.
- Our mechanisms enables users to select sensing tasks that offer the highest net profit margin based on a greedy algorithm.
- The total operation time of our approach is reduced compared with that of the task-reverse-auction incentive mechanism. The results highlight this claim and provide solutions for crowdsensing given a target area within a limited budget.

The rest of this discussion is organized as follows: survey the related work, introduce the research problem and its model, present our incentive mechanism solutions, construct simulations for evaluating the incentive mechanisms, discuss the evaluation results, consider augmented machine learning techniques, and conclude with final statements.

4.6.2 Incentive Mechanisms

4.6.2.1 Existing Monetary Incentive Mechanisms

Zhang [763] and Jaimes [299] both assort incentive mechanisms by the types of incentives. In Jaimes [299], monetary and non-monetary incentives are compared. Non-monetary mechanisms [158, 166] rely on the continued participation of users due to intrinsic motivations. Monetary mechanisms [15, 347, 356, 380, 381, 735, 743, 762, 766] rely on the direct backing of fiat money or indirect backing of fiat money through alternative currencies. According to a survey paper [763], monetary incentives will be more likely to motivate users to complete the sensing tasks than non-monetary incentives. Therefore, a monetary mechanism is more fitting for crowdsensing and will be considered in our discussion. However, picking the correct monetary incentive mechanism poses additional challenges.

When deciding on which monetary mechanism to use, one must define the optimization criteria that best suits the crowdsensing scheme. Examples of such criteria include economic feasibility, area coverage, data quality, fairness, and time duration. Producing a platform that optimizes one or more of these categories is non-trivial because the typical problem framework of a crowdsensing scheme comes down to exponential time complexity problems or typical game theoretical model challenges. In the case of economic feasibility, a platform must be designed carefully in order not to allow users too much control over the price of their service. This is known as the cost explosion problem, and is one of many challenges that must be addressed when considering appropriate monetary mechanisms. The following three monetary incentive mechanisms are well studied solutions to various crowdsensing schemes.

- The task-reverse-auction incentive mechanisms [738, 762, 766] allows for a set of users to bid on the set of tasks posted by the platform. Each bid represents a promise to finish a task provided that the platform will pay the user the bid value. Naturally, the user who bids the lowest price wins and gets the opportunity to perform the sensing task.
- In the case of the data-reverse-auction incentive mechanisms [356, 381], a set of users auction their sensing data for the posted set of tasks and their prices per already finished tasks. Then, the platform selects the data that satisfies its criteria and pays the users their bid price.
- The platform-centric model [743] treats the crowdsensing problem as a Stackelberg game. The reward of the task is changed until the platform and users reach a Nash equilibrium.

4.6.2.2 Examination of Three Typical Incentive Mechanisms

Most of the existing incentive mechanisms fall under game theoretical models. In this subsection, we examine the three aforementioned mechanisms.

The main problem of the task-reverse-auction approach [738, 762, 766] is that, because of untruthful bids, the auction style does not always select the nearest user to complete the sensing tasks [766]. In this situation, the user who is far away from a sensing task can win the auction. Further distances result in a longer travel time for users. Thus, the task-reverse-auction incentive mechanisms need more time to complete all the sensing tasks than our incentive mechanisms.

For data-reverse-auction incentive mechanisms [356, 381], while multiple users collect the data for one sensing task, only one user's data can be accepted by the platform. In other words, other users' data is wasted. As a result, this type of incentive mechanisms increases costs for car fuel, personal free time, etc. For our incentive mechanisms, users can select the sensing task before they go to collect the data. Thus, the cost explosion problem can be avoided.

The platform-centric model [743] assumes that the platform has no upper bound on budget budget. Therefore, it can find an optimal solution to giving the platform the

highest-quality data available. In practice, the platform usually has a limited budget and may not be able to obtain the game theory equilibrium reward. Our incentive mechanisms provide heuristics to obtain data while working under tighter budget constraints, and can be considered more consistent with an online crowdsensing scheme.

4.6.3 Incentive Mechanism Research Problem and Its Model

For our research problem, the platform needs to motivate the users of the platform to collect the road pavement data constraint to a budget and target area. In this case, our research objective is to design an appropriate incentive mechanism to help the platform achieve an area coverage target with a low cost and total operation time. Based on the comparison results of incentive mechanisms, the platform can choose the best incentive mechanism with the lowest budget for different area coverage targets.

Our model of the research problem contains three entities: the environment, the sensing task, and the user. Each entity can be described by its behavior and/or its relationship with other entities:

- The environment entity based on the Manhattan model; it is a grid of cells without loss of generality for incentive mechanism studies. The grid has a uniform cost distribution for traveling across adjacent cells, and no missing cells within. The environment represents the types of roads that users may encounter and the varying costs of traveling with different pavement conditions. Lastly, user entities can transfer their position only to one orthogonal cell per unit of time; users cannot move diagonally.
- The sensing task entity contains information on the location of interest and the monetary incentive associated with user participation. The sensing tasks specify roads where pavement sensing is needed.
- The user entity represents users participating in the crowdsensing scheme. As users continue to collect and report data for rewards, they accumulate monetary rewards and endure operation costs.

4.6.4 Incentive Mechanism Solutions

Modularity and scalability are critical features needed in designing a crowdsensing framework for deploying and testing incentive mechanisms. It would be difficult to swap incentive mechanisms and evaluate them without these features. Our crowdsensing platform and incentive mechanism designs are guided by the evaluation metrics described in this section.

4.6.4.1 Notations

The symbols we use in this report are shown in Table 4.6. Two important variables in our model are s_j and u_i. They represent identification numbers of the sensing tasks and users. The tasks s_j and the users u_i have attributes $< u_i, R_{ij}, x_j, y_j >$ and $< s_j, a_i, C_{ij}, x_i, y_i >$, respectively. For users, if s_j is 0 or -1, then the user is currently not participating because the user has not selected a sensing task or has dropped out. For sensing tasks, if u_i is 0 then the sensing task has not been assigned to a user. In addition, if a sensing task has a reward equal to 0, then its reward has been claimed.

Tab. 4.6: Common symbols.

Symbols	Meanings
a_i	Accumulated reward of user u_i
Avg_j	Average distance from task s_j to all users
B	Budget for the platform
BR	Base reward
b	The side length of the grid
C_{ij}	The travel cost for u_i to complete s_j
CR	The reward of the task that offers MP
$d_{j,uc}$	Distance from s_j to uc
$d_{j,tc}$	Distance from s_j to tc
IM	Incentive mechanism
k_i	The ranking number for u_i
MP	Maximum profit for user u_i
NPM	Net profit margin
P	Area coverage percentage
P_{ij}	Profit for u_i of sensing task s_j
PC	The platform cost
R_{ij}	Reward of the sensing task s_j for user u_i
(S) s_j	(Set of) Sensing task/ID
S_a	The set of available tasks
SID	The index of task selected by user u_i
s_r	The percentage of trials succeed
T	Threshold for net profit margin
tc	The center of locations of sensing tasks
t_f	Total operation time
(U) u_i	(Set of) User/ID
uc	The center of locations of users
x_i	x-coordinate
y_i	y-coordinate

4.6.4.2 Evaluation Metrics

The purpose of the evaluation metrics is to differentiate the incentive mechanisms and to guide the design of the crowdsensing solutions. The simulations for incentive mechanism evaluations consist of an extensive number of trials. In each trial, we initialize the tasks and users at the beginning and the simulation runs until all tasks are completed or all users drop out. The details of the evaluation metrics are described as follows:

- The total operation time t_f represents the duration of a trial. In one trial, a timer starts from time 0 and ends at the time t_f when all sensing tasks are completed or all users drop out. While two incentive mechanisms may have an equal success rate s_r, one incentive mechanism might have less total operation time t_f. This implies that users have been incentivized to select and perform tasks efficiently.
- The platform cost in Equation 4.12 is the amount of money that the platform pays the users through sensing task rewards. The *surplus* is the portion of the budget that is not used by the end of a trial. A lower platform cost reflects the ability of incentive mechanisms to reduce the cost of sensing task rewards.

$$PC = B - surplus. \tag{4.12}$$

4.6.4.3 Platform-Driven Greedy Algorithm

The platform-driven greedy algorithm that we use to design our incentive mechanisms is shown in Algorithm 4. The idea of this algorithm is to select an available task that gives the maximum profit to the user. Thus, this platform-driven greedy algorithm computes the gain of task s_j to user u_i by Equation 4.13.

$$P_{ij} = R_{ij} - C_{ij} \tag{4.13}$$

in which R_{ij} is determined by the incentive mechanisms. We will describe more details of R_{ij} in the following subsection. After this algorithm finds out the task s_i which can provide the maximum profit for user u_i, the user u_i needs to check if the net profit margin of the task s_i is greater than the threshold T. If positive, the user u_i selects the task; otherwise, the user u_i drops out.

4.6.5 Incentive Mechanisms

We will cover nine unique incentive mechanisms, each with unique characteristics. The task-reverse-auction (*TRA*) incentive mechanism has been discussed in the literature [738, 762, 766]. It is known that the task-reverse-auction incentive mechanism cannot guarantee that all tasks are completed within a short total operation time in untruthful bid scenarios [766]. Our incentive mechanism design has a goal of reducing the total operation time. Thus, we will compare their total operation times in Section VI. The other eight incentive mechanisms are described as follows.

Algorithm 4: Platform-driven greedy algorithm

Input: u_i, S_a, T where $u_i =< s_j, a_i, C_{ij}, x_i, y_i >$
Output: Updated $u_i.s_j$

if $S_a == \emptyset$ **then**
  $u_i.s_j = -1$; // user u_i drops out as no task is available
  return;
end
$MP = -\infty$, $CR = -\infty$;
for s_j *in* S_a **do**
  $P_{ij} = R_{ij} - C_{ij}$;
  if $P_{ij} \geq MP$ **then**
    $MP = P_{ij}$;
    $CR = R_{ij}$;
    $s = s_j$;
  end
end
if $u_i.a_i == 0$ **then**
  $NPM = 100 \times \frac{MP+CR}{CR}$;
else
  $NPM = 100 \times \frac{MP+u_i.a_i}{u_i.a_i}$;
end
if $NPM < T$ **then**
  $u_i.s_j = -1$ // u_i drops out as no task gives ample profit;
  return;
end
$u_i.a_i = u_i.a_i + CR$;
$u_i.s_j = s$;
return;

Static Uniform (SU) Incentive Mechanism In the static uniform incentive mechanism [524], the incentives of sensing tasks are fixed values that are uniformly distributed and have the value R_{ij} calculated by Equation 4.14. In this case, R_{ij} is set to the base reward BR.

$$R_{ij} = \frac{B}{|S|} = BR \tag{4.14}$$

Dynamic Relative (DR) Incentive Mechanism The incentives change their values R_{ij} based on the distance from currently unavailable users and the user u_i to the sensing task s_j. This incentive mechanism ranks the currently unavailable users and user u_i by their distance to the sensing task s_j in an increasing order. Then, the value of incentive for the sensing task s_j can be calculated by Equation 4.15.

$$R_{ij} = \begin{cases} BR & k_i = 1 \\ BR(1 - \frac{1}{2}\frac{k_i}{|U|}) & k_i \geq 2 \end{cases} \tag{4.15}$$

Dynamic/Static User-Centric (DUC/SUC) Incentive Mechanisms First, the center of user locations is calculated by Equation 4.16. Then we compute the distance $d_{s,uc}$ from the task s to the user center using Equation 4.17. The value R_{ij} is inversely proportional to the distance as shown in Equation 4.18.

- Static case: rewards of sensing tasks are computed only once at the beginning of each trial.
- Dynamic case: like the static case, but the calculation repeats whenever a user is about to select a sensing task.

$$(x_{uc}, y_{uc}) = (\frac{\sum_{i \in U} x_i}{|U|}, \frac{\sum_{i \in U} y_i}{|U|}) \tag{4.16}$$

$$d_{s,uc} = |x_s - x_{uc}| + |y_s - y_{uc}| \tag{4.17}$$

$$R_{ij} = BR(1 - \frac{1}{2}\frac{d_{s,uc}}{b * 2}) \tag{4.18}$$

Dynamic/Static Task-Centric (DTC/STC) Incentive Mechanisms This mechanism first computes the center of the locations of sensing tasks, i.e. tc, by Equation 4.19. It then calculates the distance $d_{s,tc}$ from the sensing task s to the sensing task center by Equation 4.20. Finally it derives the value R_{ij} by Equation 4.21, which is inversely proportional to the distance.

- Static case: rewards of sensing tasks are computed only once at the beginning of each trial.
- Dynamic case: like the static case, but the calculation repeats whenever a user is about to select a sensing task.

$$(x_{tc}, y_{tc}) = (\frac{\sum_{s \in S} x_s}{|S|}, \frac{\sum_{s \in S} y_s}{|S|}) \tag{4.19}$$

$$d_{s,tc} = |x_s - x_{tc}| + |y_s - y_{tc}| \tag{4.20}$$

$$R_{ij} = BR(1 - \frac{1}{2}\frac{d_{s,tc}}{b * 2}) \tag{4.21}$$

Dynamic/Static Pit (DPIT/SPIT) Incentive Mechanisms In this pit-based incentive mechanism, we use all the users' coordinates to calculate an average distance to the sensing task s by Equation 4.22. Then, we compute the incentive R_{ij} of the sensing task s by Equation 4.23.

- Static case: rewards of sensing tasks are computed only once at the beginning of each trial.
- Dynamic case: we need to recalculate the incentives when a user is about to select a sensing task.

$$avg_s = \frac{\sum_i (|x_s - x_i| + |y_s - y_i|)}{|U|} \tag{4.22}$$

$$R_{ij} = \frac{BR}{2}(1 + \frac{avg_s}{b * 2}) \tag{4.23}$$

4.6.6 Incentive Mechanism Simulation

This section describes the parameters and the processes that have been used in the simulations for the performance study of incentive mechanisms.

4.6.6.1 Parameters

The parameter tuple for each trial is $< B, P, IM >$. After simulations, the evaluation metric tuple $< t_f, PC >$ will be averaged across the total number of simulation trials. In our simulation, the unit of time and money are time unit and fiat unit. Here is the description of the parameters of the experiments:

- Budget B represents the quantity of money that allows the platform to use in a trial. For this experiment, 100 data points were collected in the interval $B \in [100.00, 1090.00]$ with 10.00 spacing between each data point.
- Area coverage percentage P represents the percentage of the area that requires sensing data. As with the budget, 100 trials were conducted such that $P \in [20.0\,\%, 79.4\,\%]$ with 0.6 % spacing between each data point. This interval represents a wide range of possible target percentages for pavement crowdsensing. Note that we round down the area coverage percentage when calculating the number of tasks.
- The final parameter is the incentive mechanism IM used in the trial. The different IM calculate rewards of tasks differently.

4.6.6.2 Simulation Execution

Given < B, P, IM >, the construction phase initializes the numbers of cells, users, and sensing tasks in the following order:

- For each cell, any references to users or sensing tasks are cleared.
- For all users, s_j, a_i, C_{ij}, x_i, and y_i are initialized. s_j is set to 0. Each user would be placed in a cell randomly without overlapping.
- For all sensing tasks, u_i, R_{ij}, x_j, and y_j are initialized. u_i is set to 0. Each sensing task will be randomly placed in a cell with no overlap between other sensing tasks.

In the execution phase, available users start their turns by selecting and committing to a sensing task based on Algorithm 4. Then, the user will update its s_j. In turn, the user information associated with the sensing task s_j will be updated to reflect that the user u_i now performs task s_j. If no suitable sensing task is found, then the user drops out of the trial for all future turns. Unavailable users are the ones who have not dropped out and commit their turns by moving towards their sensing tasks. If the user lands on the sensing task, then a_i increases by R_{ij}. If the user is not on the sensing task, then the user must wait another turn to move closer. In both cases, C_{ij}, x_i, and y_i are updated to reflect the current user location.

4.6.7 Incentive Mechanism Evaluation Results

Incentive mechanisms are evaluated and compared in three scenarios corresponding to low, medium, and high area coverage percentages for pavement crowdsensing with different numbers of users. The platform cost is used to order the incentive mechanisms based on their performance data, as shown in the following figures. The minimal budgets shown in the figures are the lowest budgets that can realize a 100 % success rate for the targeted area coverage percentage. It means that any budgets higher than this value allow the platform to achieve a 100 % success rate for the targeted area coverage.

4.6.7.1 Platform Cost Comparison

In this section, we discuss the comparison of incentive mechanisms in terms of the platform cost.

- Given 25 % area coverage, Figure 4.38 shows that the SU and DR incentive mechanisms have the lowest platform costs when the platform has 3 users and 15 users, respectively. Apart from this, the static and dynamic pit incentive mechanisms consistently rank among the top three incentive mechanisms in all scenarios.
- Given 50 % area coverage, Figure 4.39 shows that static and dynamic pit incentive mechanisms still have the best performances of the platform cost in all scenarios. Even though the DTC incentive mechanism achieves the lowest platform cost

in *50 % area coverage with 45 users*, this observation does not conflict with the previous statement.

– Given 75 % area coverage, Figure 4.40 shows that SPIT and DPIT always have the lowest platform cost regardless of how many users the platform has.

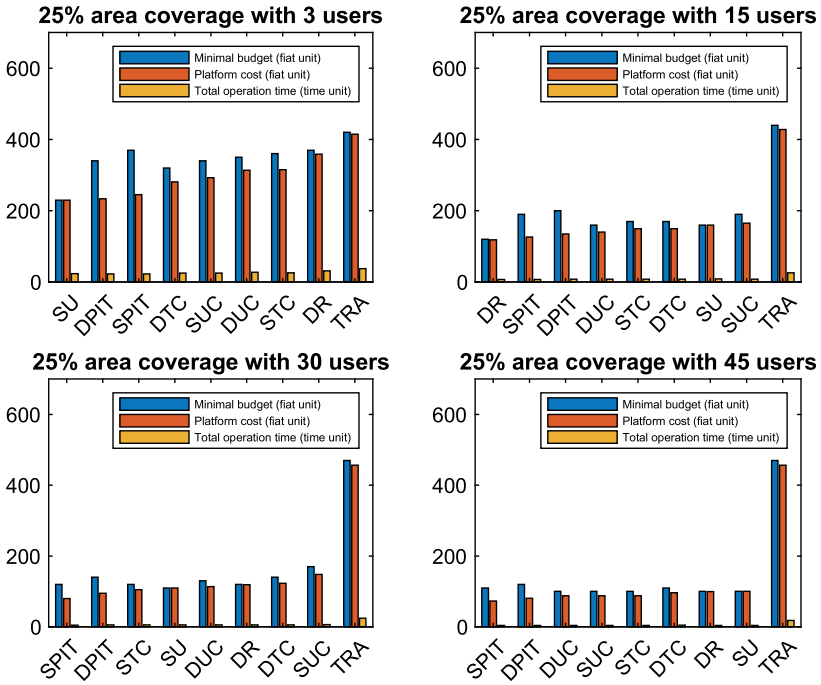


Fig. 4.38: Incentive mechanism comparison: 25 % area coverage.

Based on the observations described above, we can conclude that SPIT and DPIT are the two incentive mechanisms with the lowest platform cost.

4.6.7.2 Total Operation Time Comparison

In this subsection, we discuss the comparison of incentive mechanisms in terms of the total operation time. From Figs. 4.38, 4.39, and 4.40, the total operation time of the task-reverse-auction (TRA) incentive mechanism is nearly twice the total operation times of ours. Additionally, the total operation time of the TRA incentive mechanism becomes longer as the number of participatory users increases while the total operation times of our incentive mechanisms would decrease in the same situation. This result proves that our incentive mechanisms have much less total operation time than the Task-Reverse-Auction (TRA) incentive mechanism.

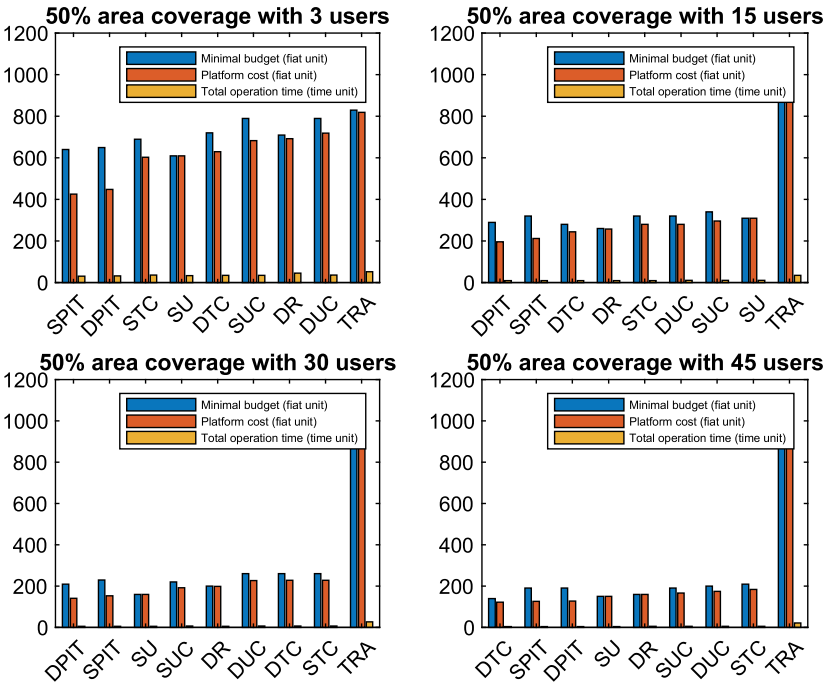


Fig. 4.39: Incentive mechanism comparison: 50 % area coverage.

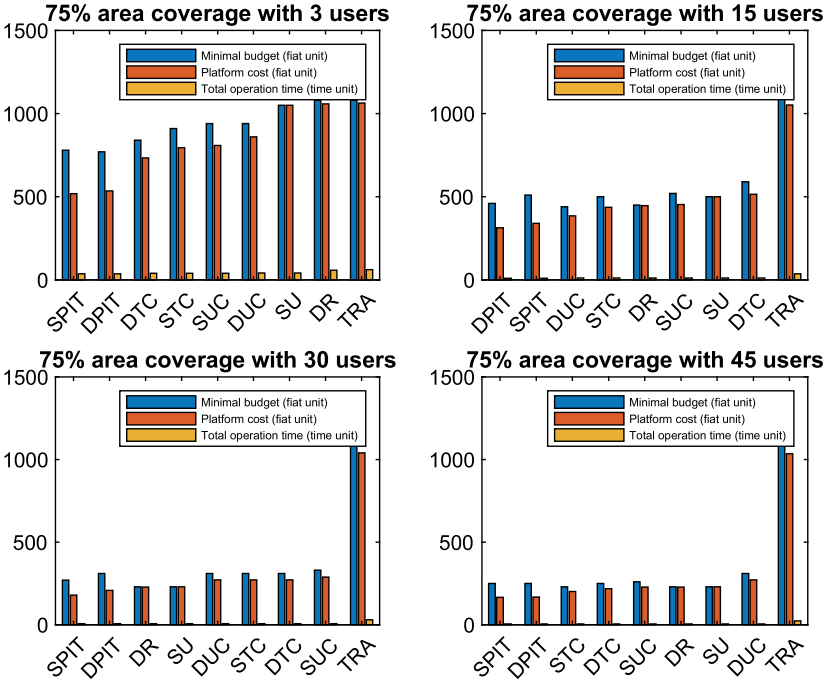


Fig. 4.40: Incentive mechanism comparison: 75 % area coverage.

4.6.8 Machine Learning Augmentation

Based on computer vision, crowdsensing and machine learning can be applied to pavement distress monitoring. For example, we have used a state-of-the-art machine learning model for detecting pavement damages based on images captured by the Android-phone camera and classifying them into eight types with corresponding confidence [738]. The types include (i) Liner crack, longitudinal, wheel mark part, (ii) Liner crack, longitudinal, construction joint part, (iii) Liner crack, lateral, equal interval, (iv) Liner crack, lateral, construction joint part, (v) Alligator crack, (vi) Rutting, bump, pothole, separation, (vii) White line blur, and (viii) Crosswalk blur. We have chosen this machine learning model because it "achieved recalls and precisions greater than 75 % with an inference time of 1.5s on a smartphone." [738]

Moreover, machine learning has been used for incentive mechanisms in embedded crowdsensing applications. For example, neural network and clustering algorithm have been applied for user grouping in [427] and the resulting incentive mechanisms can reduce the social cost, overpayment ratio, and grouping time. In the following sections, we discuss general considerations of machine learning augmentation, supervised learning, and unsupervised learning for incentive mechanisms.

4.6.8.1 General Considerations

We have witnessed great strides in the development of machine learning and its applications in recent years. Work in the spaces of image classification, text generation, language translation, and generative adversarial networks have produced results that could only be described as magic to the untrained eyes. Furthermore, being able to harness the full potential of these techniques, including subsequent derivatives, will be the aim of research for the foreseeable future. Such tantalizing thoughts act as motivation to incorporate different machine learning algorithms within the framework of crowdsensing, and our discussion is no exception. The crowdsensing scheme can be augmented using both supervised and unsupervised learning.

- Supervised learning focuses on mapping input data to target classes. Typically the input data will be sampled from a dataset where particular data points must be categorized. One common supervised learning task is image recognition. For example, one may use a convolutional neural network that receives input data in the form of road pictures and produces output data in a string distinguishing the road condition [738].
- Unsupervised learning focuses on clustering datasets such that embedded classes may be revealed. These class embeddings may reveal subsets of data and help highlight underlying relationships introspectively. One common unsupervised learning task is dimensionality reduction. In this case, we may consider a dataset with copious amounts of features. Using a clustering algorithm, such as spectral clustering, we may be able to reduce the principal features required to character-

ize data points uniquely. In other words, we may only need a proper subset of a dataset's features to represent the grouping of a data point. These transformations may lead to lossless decomposition of our data. Thus, we can reduce memory stress for supervised learning tasks.

4.6.8.2 Supervised Learning for Predicted Budget

One apparent shortcoming between all covered works in this contribution is the predicted budget. Having incentive mechanisms work under a limited budget is only reasonable if the budget has been methodically selected. In this case, a supervised learning problem is clearly established. Either a classification of the simulation parameters or a forecast of user costs can be used to determine a predicted budget. In the first case, the input data would include basic simulation parameters, such as the number of users and percentage area coverage, and the output would be the predicted budget. In the second case, the simulation would provide a seed state, including coordinates of users and sensing tasks, as input and calculate the final board statistics in terms of overall operation costs. The overall operation costs would be correlated with the predicted budget. Previous simulations would act as the data needed to construct these models in either case.

4.6.8.3 Unsupervised Learning for Incentive Mechanisms

Although the different incentive mechanisms studied in this contribution showed various levels of effectiveness in finishing the sensing tasks, a question of interest in crowdsensing applications is how their performances might differ if the incentives follow a non-uniform or random distribution. One may model the environment of the crowdsensing scheme as scattered normal distributions where sensing tasks may cluster in different neighborhoods. This scenario is realistic in rural living where communities may be sparse apart but dense around some centroid. The covered incentive mechanisms require a new component to scale incentives across clustered communities effectively. A clustering algorithm could locate the centroid of sensing task clusters and calculate rewards relative to these neighborhoods for the unsupervised learning task. In this case, the incentive mechanisms could scale to any size of environment given sufficient resources.

4.6.9 Conclusion and Future Work

In this contribution, we proposed eight incentive mechanisms based on a platform-driven greedy algorithm to help the crowdsensing platform motivate users to collect pavement condition data. Since our incentive mechanisms allow users to select the sensing tasks based on a platform-driven greedy algorithm before they start to collect the data, they can avoid the cost explosion problem observed in the data-reverse-

auction incentive mechanisms. From the simulation results, we find that SPIT and DPIT are the incentive mechanisms that have the lowest platform cost. Compared with the task-reverse-auction incentive mechanism, our incentive mechanisms reduce the total operation time by half. Our future research includes large-scale simulations and real-life experiments by extending our prototype pavement crowdsensing system. Lastly, we discussed machine learning augmentations for embedded crowdsensing applications and different incentive mechanisms.

4.6.10 Acknowledgment

The contents of this section reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The project is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. government assumes no liability for the contents or use thereof. This work was sponsored by a grant from the Center for Integrated Asset Management for Multimodal Transportation Infrastructure Systems (CIAMTIS), a U.S. Department of Transportation University Transportation Center, under federal grant number 69A3551847103, and the Commonwealth of Pennsylvania's PITA program. The authors are grateful for the support.

5 Communication Networks

5.1 Capacity Analysis of IoT Networks in the Unlicensed Spectrum

Stefan Böcker
Christian Arendt
Christian Wietfeld

Abstract: The ongoing digitalization and the steadily increasing number of distributed sensor devices and Internet-of-Things (IoT) systems implies a massive increase of subscribers. At the same time, the amount of available frequency spectrum resources remains static. In this respect, current 5G networks are already aiming for large-scale connectivity with an ambitious node density of 1 000 000 devices per square kilometer in the area of *massive Machine Type Communication* (mMTC). A huge number of potential technology solutions are available, but a comprehensive networking solution based on one technology seems unlikely. Among typical cellular IoT technologies, these challenging 5G mMTC requirements are also addressed by a growing number of unlicensed technologies enabling a simple, cost-effective network operation independent of licensed operators.

In this context, the potentials of Low-Power Wide Area Networks (LPWAN) technologies, as an additional technology option in unlicensed frequency bands are analyzed. Specifically, this work aims to analyze the suitability of LoRaWAN to contribute to given 5G requirements for specific mMTC applications in large-scale deployments. The performance evaluation illustrates that LoRaWAN is attractive due to high communication ranges up to multiple kilometers, enabling a high coverage even with a small number of cells. The evaluation also finds that the technology has a high potential to contribute to 5G mMTC application areas, especially for non-time-critical sensor applications.

To further increase the reliability of LPWAN systems, especially for critical services, different approaches to increase spectral efficiency are discussed. In addition to purely scheduling-based approaches, a data-driven analysis of the spectral power density to predict and avoid technology-independent interferences is presented. This is used to increase the robustness of LPWAN systems by centrally deriving communication profiles that address and bypass the predicted interference characteristics.

Apart from intelligently scheduling data transmission, another way for increasing efficiency is to reduce the amount of data that has to be transferred in the first place. Nowadays, initial generations of connected IoT devices and applications enabled by Cellular-IoT (CIoT) and LPWAN technologies are deliberately kept simple and based on equidistant, regular communication intervals. By contrast, we illustrate an Artificial

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.
https://doi.org/10.1515/9783110785982-005

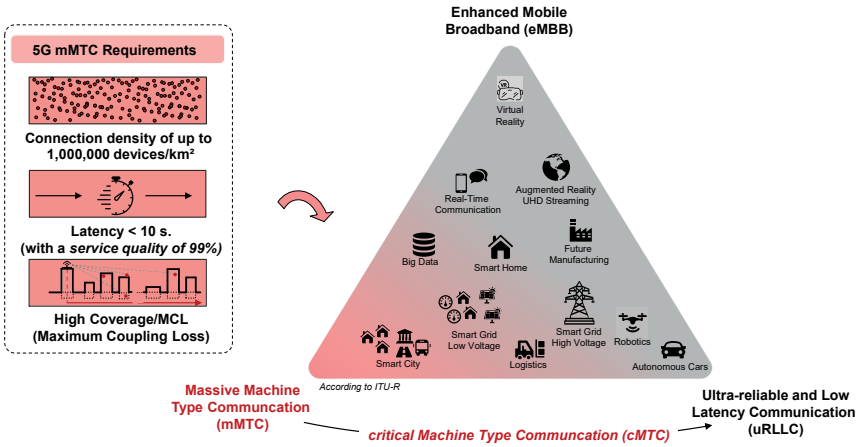


Fig. 5.1: Achieving the 5G scalability targets for IoT environments

Intelligence (AI)-based model-predictive communication approach taking advantage of knowledge about the underlying data of a sensor system. An AutoRegressive Integrated Moving Average (ARIMA) model is used in order to depict the behavior of an applications sensor data, leaving only values deviating from the model to be transmitted.

5.1.1 Introduction

Wireless connectivity has become a ubiquitous part of daily life. While wireless networks were originally developed to connect people, cellular networks have evolved to enable Machine-to-Machine (M2M) communication. In a wide variety of application areas, such as smart cities, energy systems, or production and logistics, devices are linked to each other to enable fully autonomous operation without human intervention. In this context, the 5G specification defines an mMTC requirement profile that aims for an ambitious scalability target of 1 million subscribers per square kilometer, while maintaining a maximum latency of 10 s (see Figure 5.1). At the same time this scalability target is linked to a boundary condition, that considers a pre-defined Poisson arrival process traffic pattern for non-full buffer systems with a payload of 32 B.

While Section 4.3 presents the performance evaluation of the NB-IoT technology as a current 3GPP solution to address 5G mMTC requirements, this contribution covers the research challenge to identify complementary technologies operated in unlicensed frequency bands to contribute to tight 5G mMTC requirement profiles. To this end, this work first discusses potentials and limitations of the Long-Range Wide-Area Network (LoRaWAN) technology, as a representative of LPWAN solutions, in order to subse-

quently introduce optimizations for further performance enhancements of unlicensed technology solutions with specific respect to mMTC applications.

5.1.2 Opportunities and Challenges of the Licensed and the Unlicensed Spectrum for IoT Environments

First, this section introduces the necessary LoRaWAN fundamentals from which the opportunities and challenges are derived. In addition, the impact of regulatory policies on the performance of the LoRaWAN technology is discussed.

LoRaWAN is an LPWAN specification for wireless battery-powered systems in a regional, national, or global range. It is based on the LoRa Modulation technique and mainly operated in the Short-Range Device (SRD) band at around 868 MHz in Europe and 915 MHz in the US. LoRaWAN enables a wide-range communication even in urban scenarios and provides a very good deep indoor penetration [203],[304]. The definition of the spreading factor (SF) permits the trade-off between efficient and very robust communication, whereby the data rates vary from 0.25 kbit/s (SF=12) to 5.5 kbit/s (SF=7).

Because LoRaWAN is operated in unlicensed frequency bands, the channel access must comply with regulatory frequency band requirements that ensure that all participants have equal access to frequency resources. For the underlying SRD band, the European Commission in cooperation with ETSI allows mitigation techniques such as Listen Before Talk (LBT), detect and avoid (DAA), and duty-cycle limitations [192], whereby LoRaWAN relies on a fairly simple pure ALOHA channel access and implements the duty cycle limitations to meet regulatory ETSI requirements. Thus, peak physical data rates are further reduced by a factor of more than 99 % due to the regulatory impact of a given duty cycle of 1 % and a MAC overhead with minor impact. The resulting average throughput ranges from 1.5 to 48 bit/s (as shown in Figure 5.2).

Consequently, throughput limitations are mostly driven by the idle time (time off) following the transmission time per packet (time-on-air), which is required to meet duty cycle limits [446]. The LoRaWAN specification defines three mandatory channels: 868.1 MHz, 868.3 MHz, and 868.5 MHz, additional resources are optional (see Figure 5.3). To reduce interference, channels are cycled in a pseudo-random approach.

As shown in Figure 5.3, the duty-cycle constraints apply to each SRD sub-band and may vary between different sub-bands, i.e. a dedicated LoRaWAN downlink communication is deployed at 869.525 MHz with a duty cycle of 10 % and additional limiting to data-rate class 0 (DR0).

5.1.3 Capacity Limits of LPWAN Technologies in Unlicensed Band Operation

The determination of capacity limits is based on a performance evaluation derived from an analytical model [446] that has been enhanced fundamentally for the underlying

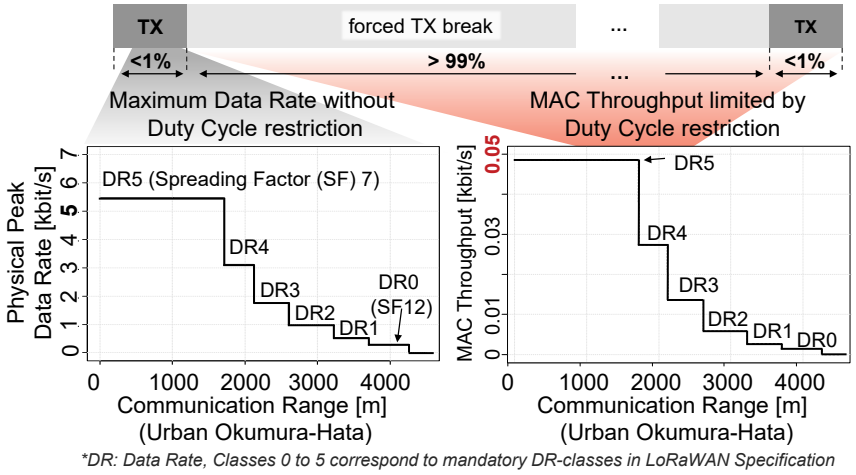


Fig. 5.2: Impact of regulatory limitations in unlicensed band deployments ©[2019] IEEE. Reprinted, with permission, from [93].

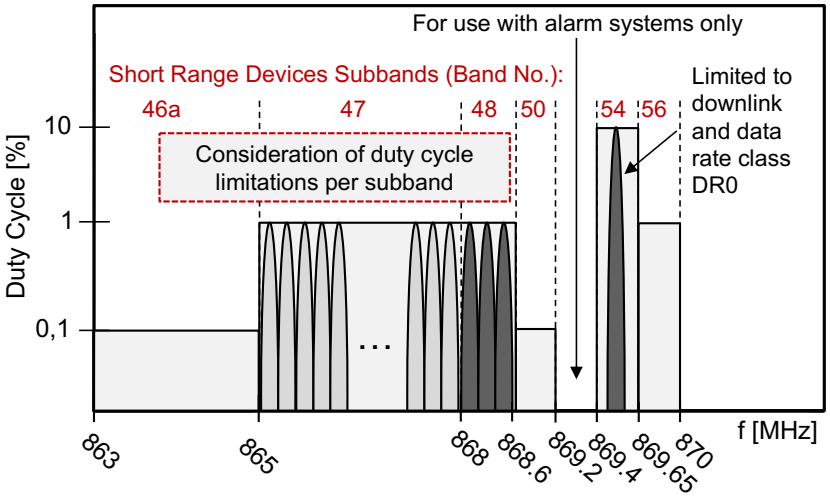


Fig. 5.3: European LoRaWAN channel frequencies in the ISM Band (EU863-870) ©[2019] IEEE. Reprinted, with permission, from [93].

work. The resulting analytical model enables demand-based derivation of key performance parameters, such as data rate and coverage area. The main extension aims at deriving latency bounds for different scalability scenarios and large-scale deployments, enabling the simultaneous determination of service guarantees.

Because downlink communication between a LoRaWAN gateway and distributed LoRaWAN nodes is not interfered with by uplink communication, performance limits

can be easily determined by evaluating a maximum capacity of a single LoRaWAN link. Considering the mandatory duty-cycle constraint (see Section 5.1.2), this results in overall limited downlink performance margins. Based on the assumption of Class A LoRaWAN nodes, Figure 5.4 depicts the average downlink throughput of a LoRaWAN network. As illustrated, the LoRaWAN downlink for Class A devices is based on two consecutive downlink receive windows. Following an uplink message, a Class A end device opens a first receive window (RX1) typically one second later. The first receive window is opened one second after termination and on the same frequency channel as the previous uplink message. The second receive window (RX2) is typically opened 2 s after uplink transmission and based on a dedicated downlink channel at 869.525 MHz with a mandatory duty cycle of 10 % and a limitation to data rate class DR0. This results in a low average downlink throughput of about 31.2 bit/s, which excludes a large number of downlink-intensive and safety-critical applications, such as update or upgrade functions.

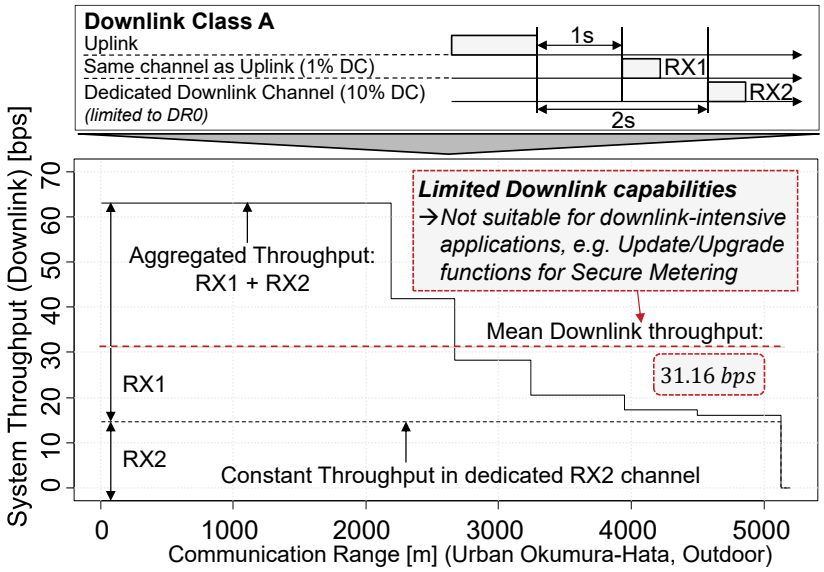


Fig. 5.4: System throughput (downlink) utilizing maximum duty-cycle capabilities ©[2019] IEEE. Reprinted, with permission, from [93].

In the uplink communication direction, the pure ALOHA channel-access scheme is implemented. In this context, the parallel uplink communications of different spreading factors are orthogonal to each other, permitting each data rate class to be modeled as an independent ALOHA process. The maximum system throughput of each data rate class is closely related to the number of devices and can be determined using the known ALOHA model equation $S = G \cdot e^{-2G}$, whereby S is the normalized channel throughput

and G the channel traffic [8]. The integration of the capture effect takes into account that two parallel transmissions with an Received Signal Strength Indicator (RSSI) delta greater than 6 dB for co-channel rejection [592] do not interfere with each other. In this case, only one packet is dropped, and the packet wirh higher received power is assumed to be decoded successfully, resulting in increased scalability. ALOHA equations can be adapted to $S = \frac{G \cdot e^{-2G}}{2} \cdot (1 + e^{G})$ [140]. In order to allow the derivation of guaranteed performance in addition to maximum uplink system throughput or scalability, the existing model is extended to incorporate the derivation of latency bounds, whereby latency is defined as the sum of Time-on-Air (ToA) and forced time off (T_{off}) by regulatory duty-cycle restrictions. Consequently, every interfered transmission increases the time on air $ToA+T_{off}$. Therefore, the mean latency $\overline{\tau_{DR}}$ can be defined based on the maximum packet collisions per transmission $\frac{G}{S}$ as depicted in the following equation:

$$\begin{aligned} \overline{\tau_{DR}} &= \frac{G}{S} \cdot (ToA + T_{off}) - T_{off} \\ &= (e^{2G} - 1) \cdot (ToA + T_{off}) + ToA \end{aligned} \tag{5.1}$$

Equation 5.1 is enhanced to consider the 99 %-quantile of defined 5G mMTC latency requirements $\tau_{99\%}$.

$$\tau_{99\%} = \log_{1-e^{-2G}}(0.01) \cdot (ToA + T_{off}) + ToA \tag{5.2}$$

The derived model can be modified and configured depending on desired parameter scenarios. Figure 5.5 illustrates the results for an exemplary configuration of a 32 B payload, 3 channels, and the maximum duty cycle. Without considering the capture effect, the maximum throughput of approximately 3.3 kbit/s can be achieved with a fleet size of about 900 subscribers. This can be further increased by about 50 % due to the additional consideration of the capture effect, resulting in a maximum throughput of about 4.7 kbit/s for a uniformly distributed number of subscribers of 1350. However, taking into account the limiting data-rate class DR0, this is simultaneously accompanied by an increased average latency of 400 s, which corresponds to an increase of about 12.5 %.

5.1.3.1 LoRaWAN Contribution to 5G

LoRaWAN technology is emerging as a very good solution in the unlicensed spectrum band to support 5G mMTC targets. Although LoRaWAN does not support the required 164 dB with a loss of 151 dB, it can cover the targeted area of one square kilometer, which is defined for the 5G mMTC connection density target. At any rate, it has a very good range and deep indoor availability even for urban areas [304]. Furthermore, Figure 5.6 illustrates the impact of various LoRaWAN parameter configurations on maximum scalability in view of 5G mMTC connection density and latency requirements (see Section 5.1.1). It can be shown that the 5G mMTC parameter set results in a significant contribution of 10 % for three 125 kHz frequency channels, which can be increased up

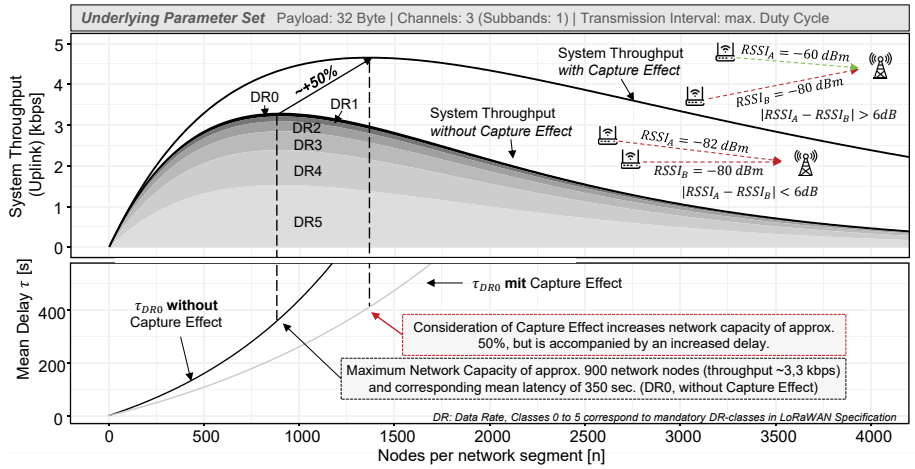


Fig. 5.5: System throughput (uplink) and latency utilizing maximum duty cycle capabilities ©[2019] IEEE. Reprinted, with permission, from [93].

to 25 % when the 8 channels are considered to cover 5G mMTC targets of one million devices per square kilometer. However, these results are obtained without considering the 5G mMTC latency requirement of 10 s. If the latency requirement is taken into account to ensure a certain quality of service, scalability is reduced by 25 % for a 50 %-quantile or up to 70 % under consideration of a 99 %-quantile.

When deviating from the 5G mMTC traffic pattern and considering other payload sizes, it can be seen that this factor has only a minor impact on scalability. By contrast, the variation of the transmission interval has a fundamental effect. In the case of a low transmission interval of only 12 hours, more than 50 % can be met of the overall connection density of one million devices per square kilometer without consideration of a latency requirement (99 %-quantile). By contrast, almost 20 % can be met when taking into account a latency requirement. So far, the results described have been focused on the assessment of unacknowledged traffic in the uplink direction. When ACK packets in the downlink direction are included, the downlink indicates a significantly constrained scalability of the LoRaWAN network. Even for a transmission interval of 12 hours, the scalability decreases to about 14 250 subscribers per square kilometer, which corresponds to a reduction of 97 % and is consistent with the limitations of the LoRaWAN downlink. Overall, depending on the application scenario and configuration, a very significant contribution of LoRaWAN technology to the 5G mMTC goals can be deduced, though the application field should be limited to non-time-critical sensor applications.

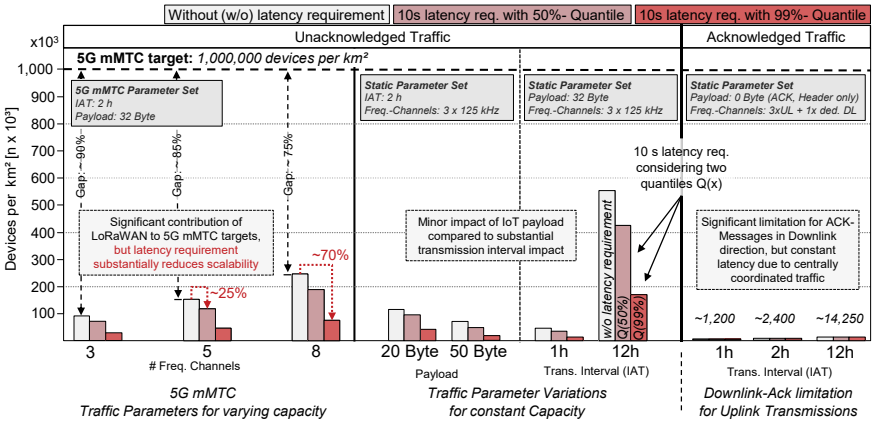


Fig. 5.6: Impact of various LoRaWAN parameter configurations on maximum scalability considering 5G mMTC connection density and latency requirements ©[2019] IEEE. Reprinted, with permission, from [93].

5.1.4 Data-Driven Capacity Improvements

In this section, optimization methods are presented that further enhance the performance of the previously discussed LPWAN technologies in unlicensed frequency bands.

5.1.4.1 Dynamic Spectrum Management to Improve Scalability of Time-Critical Sensor Applications

Due to their simple and cost-effective technical viability, a steadily increasing number of LPWAN are operated in unlicensed frequency bands. For each user, this leads to a large number of possible interference effects caused by a wide variety of technologies, each using different channel access methods. There is no central coordination as in licensed mobile radio frequencies. Despite mandatory interference mitigation techniques, quality of service in terms of availability, latency, etc. cannot be guaranteed due to uncoordinated channel access in unlicensed frequency ranges. In order to tackle these challenges, a data-driven spectrum management procedure is proposed (see Figure 5.7). This approach relies on SDR -based spectrum sensing to gather information on channel occupation, which is used to predict future spectral utilization.

The predicted activity profiles are used for spectrum management in order to intelligently schedule future transmissions. Three scheduling approaches, namely Restricted Access Window, Weighted Restricted Access Window, and Coordinated Restricted Access Window, have been developed and evaluated. An overview of these approaches is shown in Figure 5.8, a brief description of each approach is given below. For evaluation, an externally defined Key Performance Indicator (KPI) is required, which is represented by the expected latency in this work. As LoRaWAN is the technology of choice, an

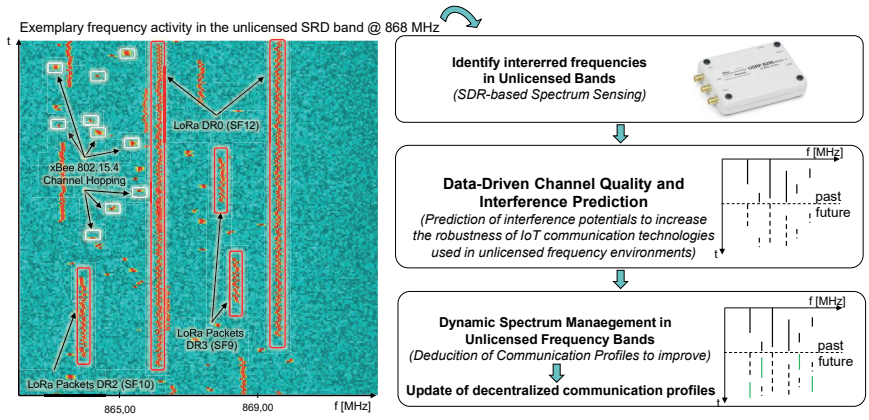


Fig. 5.7: Analyzing and reducing the impact of interference in unlicensed bands.

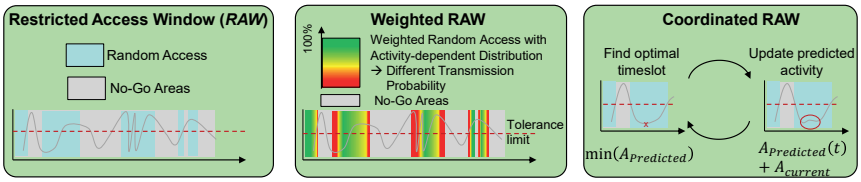


Fig. 5.8: Concept of the dynamic spectrum-management approach.

ALOHA channel access is used as the basis and a duty cycle has to be observed. The expected latency is modelled as described in Section 5.1.3

The three optimization methods Restricted Access Window, Weighted Restricted Access Window, and Coordinated Restricted Access Window are described below.

Restricted Access Window (RAW) The basic idea and the name are derived from IEEE 802.11ah technology. The time ranges are first divided into no-go areas and random access areas. In this case, the channel is accessed randomly in areas below the drawn-in threshold and access is avoided in the no-go areas. Restricted Access Window is the simplest developed method. This method is a coarse-grained method, which means that the gateway does not determine the transmission time for each subscriber; rather, the gateway only transmits the appropriate transmission time through beacons and the subscribers randomly select a transmission time.

Weighted Restricted Access Window In this method, the Restricted Access Window method is extended by a weighted access probability. This means the time ranges with an activity level below the tolerance limit have a higher chance to be selected.

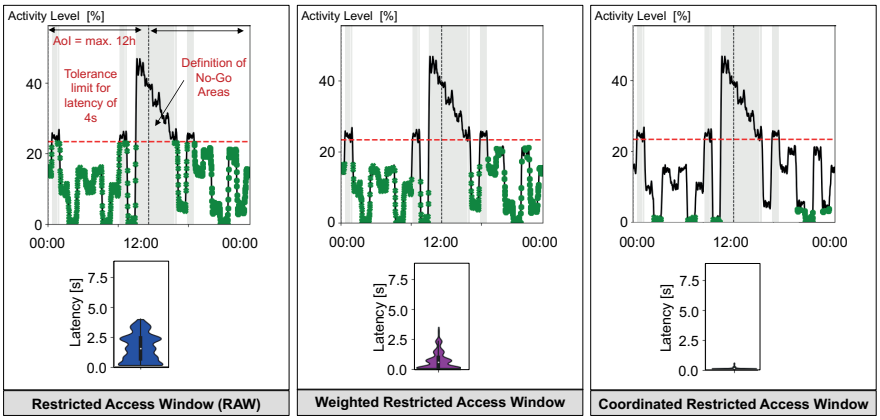


Fig. 5.9: Comparison of the three scheduling approaches with regard to potential latency reduction.

In this work, a quadratic functions is used as weighting and the weighting is calculated for each time point using Equation 5.3.

$$W_t = (A_{Tolerance} - A_t)^2 \tag{5.3}$$

This method is a coarse-grained method and the gateway transmits matching transmission time points with their weights by beacons. The participants finally choose a transmission time point by a weighted random.

Coordinated Restricted Access Window (CRAW) This procedure finds the optimal transmission time for each participant by assigning the time with the minimum activity to the first participant. Then the activity profiles are updated with the resulting activity by first participants. This procedure is repeated for the other participants.
The cyclic steps of the procedure are as follows:

1. finding the minimum activity in profile and assigning this time to the participant;
2. updating the profile in view of the activity change at the assigned time point; and
3. repeating the procedures for the next participant.

Compared with the previous methods, CRAW is a fine-grained scheduling method. This means that the gateway must communicate the transmission time to each subscriber. Therefore, this method requires many resources. In CRAW, the influence of each participant on the activity is taken into account.

Comparison of Scheduling Performance In this section, the three developed methods are evaluated using the presented scenarios. The predicted daily profiles from the 868 MHz study are used as a basis. In Figure 5.9, the results are shown and the used daily profiles are represented at the top.

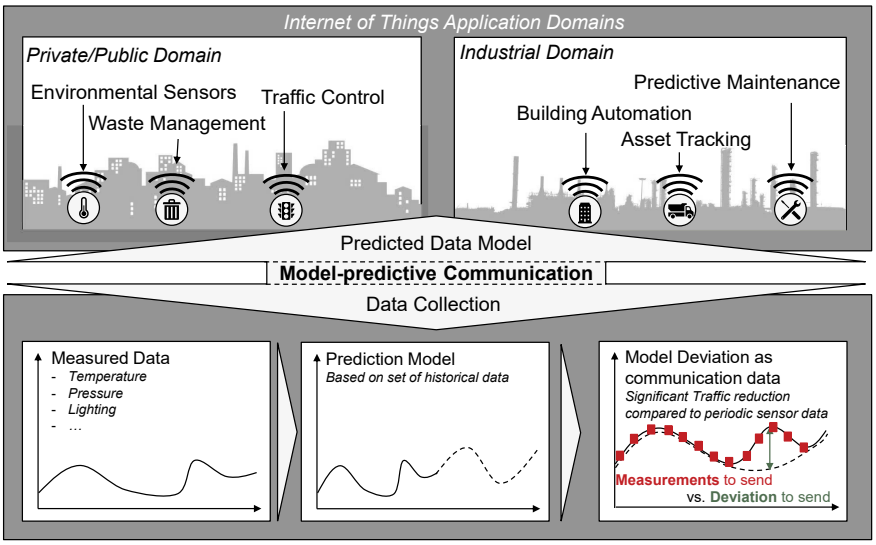


Fig. 5.10: Model-predictive communication in Internet of Things environments ©[2020] IEEE. Reprinted, with permission, from [30].

It can be seen that the expected latency can be significantly reduced by using more intelligent scheduling mechanisms, which avoids times with high channel activity. Simply avoiding regions with a mean latency over a given threshold with the RAW approach sets the baseline potential, which can be optimized using a weighting to reduce the mean latency below 1 s in this study. Providing more intelligence for the scheduling method with CRAW pushes all observed latencies below 1 s.

However, this approach generates an increased computational effort, which has to be taken into account as a trade off.

5.1.4.2 Data-Driven Model-Predictive Communication

In this section, we propose a data-driven approach to reduce communication efforts by leveraging knowledge about the underlying sensor data in IoT systems. In order to keep devices simple, the data transmission of IoT devices typically follows a regular pattern of equidistant time intervals. This leaves a high potential for optimizing the efficiency of spectral resource usage. Therefore, this study proposes a model-predictive communication framework that leverages knowledge about the underlying sensor data and allows IoT devices to rate the value of observations in order to decrease communication effort and free up spectral resources for other parties. This approach potentially increases the number of devices considered in the scheduling of resources in licensed frequency bands and reduces the likelihood of interference in unlicensed bands. Figure 5.10 depicts the concept of this work.

The underlying approach is not to transmit every measurement, only those that deviate significantly from a predetermined model. Therefore, two time series forecasting mechanisms are examined in this study to generate such a model of a temperature sensor setup. Because these methods simply rely on past data, the use case may be readily changed. This approach has been evaluated by using a dataset originating from an environmental indoor sensor located in a residential area in Dortmund, Germany. From the 1st of January 2019 until the 19th of November 2019, the system gathered temperature, humidity, and CO_2 concentration at a frequency of around 5 minutes. To minimize model complexity, the dataset was resampled to 30-minute time steps in this work. The raw data can be accessed via [29]. The forecasting algorithms used in this work are an autoregression based approach and a neural network approach, which are described in the following sections. Both models take advantage of a decomposition approach, which is discussed more below.

Seasonal and Trend Decomposition Using LOESS (STL) In this work, the Seasonal and Trend decomposition using LOESS (STL) method [540] is used to extract typical properties of the underlying data, such as a daily profile for temperature data. The algorithm consists of two loops: the inner loop uses LOcally wEighted Scatterplot Smoothing (LOESS) [141] to extract the seasonal and trend components and the outer loop is used to minimize the impact of outliers by computing robustness coefficients. An example decomposition of a daily temperature profile from the dataset is shown in Figure 5.11.

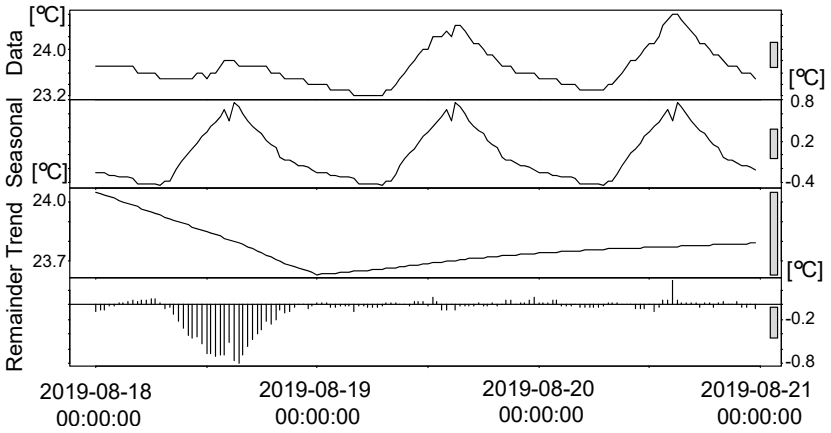


Fig. 5.11: Decomposition of measured temperature data from 18th of August until 21st of August using STL ©[2020] IEEE. Reprinted, with permission, from [30].

The example shows strong daily seasonality and a local weather trend. Additionally, a significant remainder is present that could not be related to either the trend or seasonality component. The decomposition can be used to aid forecast algorithms by subtracting the seasonal component before applying the prediction, and adding it back to the resulting time series afterwards. As the seasonal component typically changes slowly over time, it is possible to reduce the complexity of the analyzed time series in order to decrease prediction errors.

AutoRegressive Integrated Moving Average (ARIMA) The ARIMA algorithm is a state-of-the-art time series forecasting method and one of the most widely used. It is composed of three components: the autoregressive part $AR(p)$, entailing the past values of the original series; the integrated part $I(d)$, related to differencing in order to make the time series stationary; and the moving average component $MA(q)$ marking the model errors. The parameter set (p,d,q) defines the order of each component and therefore indicates the specific ARIMA model in use. In detail, p indicates the number of considered past values, d is the differencing degree and q specified the considered previous error terms. To simplify the application in this work, the forecast package for the statistical programming language R is used. The language contains an implementation of the ARIMA algorithm, which allows automatic parameter set selection for every model realization [294], where a unit test procedure checking for stationarity is used to specify d, while p and q are found by minimizing Akaike's Information Criterion (AIC) . A prediction for the 21st of August 2019 using ARIMA(0,1,0) based on a training period of three previous days together with the 95 % and 80 % prediction intervals as well as the actual measurement as test data is depicted in Figure 5.12

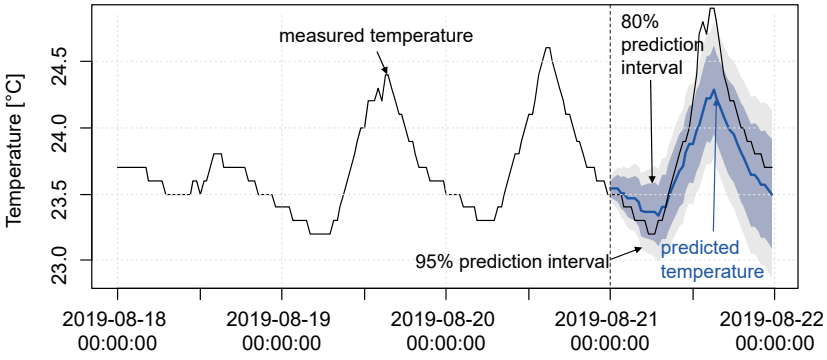


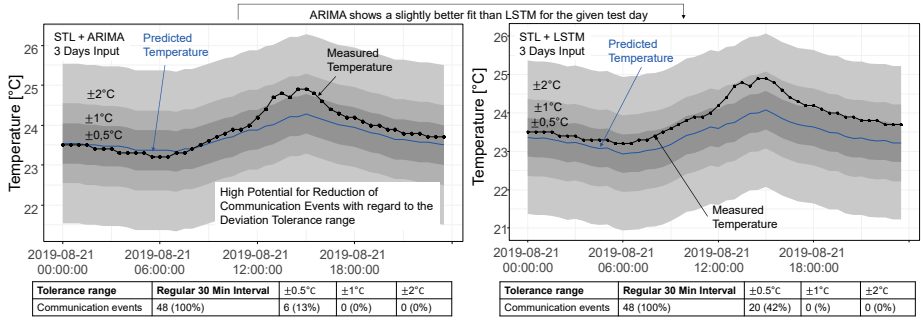
Fig. 5.12: Forecast for one sensor using STL + ARIMA(0,1,0) for the 21st of August 2019 with 95 % and 80 % prediction intervals ©[2020] IEEE. Reprinted, with permission, from [30].

It is evident that a good match is achieved between the predicted and the measured temperature, with most of the measured temperature values lying inside the 80 % prediction interval.

Long Short-Term Memory (LSTM) As a variant of Recurrent Neural Networks (RNN) developed by Hochreiter et al. [277], LSTM networks have an improved ability to learn long-term dependencies in a dataset, which makes them appropriate for time series prediction tasks. The main structure of LSTM networks consists of concatenated cells that are linked together by constant cell states and the input flow. An input gate is used to regulate the influence of the cells input, while a forget gate filters previous cell states.

The implementation used in this work is based on the well-known keras python framework with the Theano library as a backend. In order to keep the training duration managable and avoid decreasing the model accuracy by deeper networks as stated in [358], a single network layer is used. 50 LSTM cells were used within this layer, however the influence of the number of cells was seen to be minor.

Figure 5.13 compares LSTM and ARIMA performance for an exemplary prediction based on the data of the 21st of August 2019 with the three previous days as an input in terms of required communication events with different tolerance ranges of 0.5 °C, 1 °C, and 2 °C, respectively. The tolerance ranges depict the need for communication events, as only deviations higher than the tolerance should be transmitted.



Tolerance range	Regular 30 Min Interval	±0.5°C	±1°C	±2°C
Communication events	48 (100%)	6 (13%)	0 (0%)	0 (0%)

Tolerance range	Regular 30 Min Interval	±0.5°C	±1°C	±2°C
Communication events	48 (100%)	20 (42%)	0 (%)	0 (0%)

Fig. 5.13: Forecast for one sensor using STL + ARIMA(0,1,0) and LSTM for the 21nd of August 2019 with tolerance ranges of ± 0.5 °C,± 1 °C and ± 2 °C. ARIMA shows a slightly higher potential in decreasing communication effort for ± 0.5 °C tolerance. ©[2020] IEEE. Reprinted, with permission, from [30].

ARIMA can achieve a potential reduction of communication events by 87 % for this day with a tolerance of 0.5 °C, where LSTM provides a lower potential of 58 %. However, for applications where a higher tolerance of 1 °C is sufficient, both algorithms provide enough accuracy to save up to 100 % of the communication effort.

Statistical Performance Evaluation Both implemented approaches are statistically evaluated using the Root Mean Square Error (RMSE) as a metric for the modeling errors and for the potential reduction of communication events. To make both modeling concepts comparable, a sliding window approach as depicted in Figure 5.14 is used for validation. Each sliding set from the total dataset has a configurable number of input

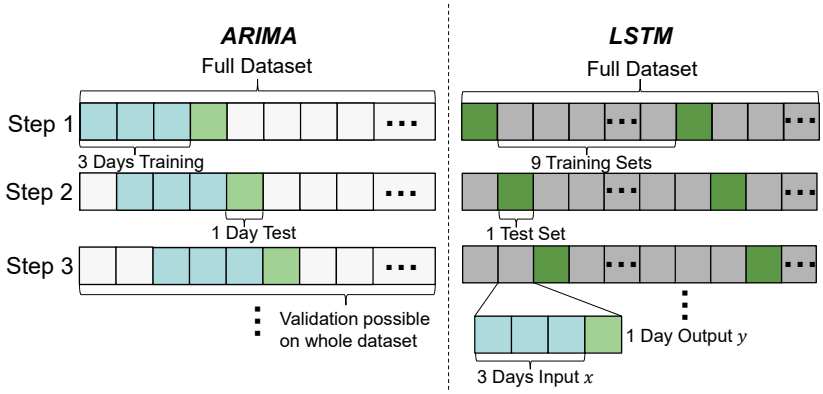


Fig. 5.14: Walk forward validation with sliding window (constant training period length) in each algorithm's variant. ARIMA takes t days of training data and produces a one day forecast that is compared to the test day, LSTM needs input/output pairs for training and is then tested on a held back test set. ©[2020] IEEE. Reprinted, with permission, from [30].

days followed by one output day. An advantage of ARIMA is the ability to sufficiently predict future data relying solely on a small data basis. LSTM, on the other hand, needs a large portion of input/output pairs as training data in order to learn essential features enabling the sensor data prediction. The impact of varying numbers of input days from 3 to 12 on the prediction of one forecast day has been addressed in this work. A 10-fold cross-validation is used to further validate the forecast results, with a 90-10 % split between training and test data. The resulting modeling error of both approaches for varying number of input days represented by the RMSE distribution is illustrated in Figure 5.15.

It can be seen that the impact of the input period length is small, but both models have a slightly lower error with smaller input periods. In general, LSTM provides a doubled mean error of around 0.3 °C when compared with ARIMA, producing a mean error of around 0.15 °C for all input period lengths. LSTM also shows a higher spread of the observed errors, except from a small amount of outliers experienced with ARIMA for longer input periods. The latter lead to the conclusion that data lying further in the past is less relevant for predicting the temperature values of the following day, increasing the amount of false forecasts observed. These results allow an estimation of the tolerance ranges in which the evaluated models can be used. A tolerance range of ±

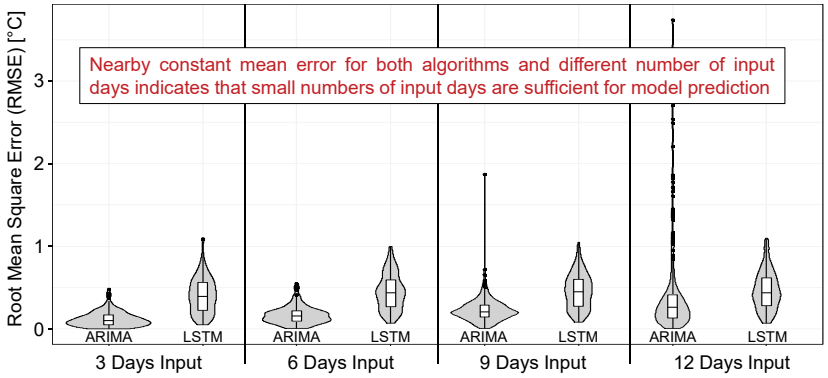


Fig. 5.15: RMSE for both approaches and varying training period. ARIMA shows a smaller RMSE than LSTM, however the mean error for both approaches remains nearly constant. ©[2020] IEEE. Reprinted, with permission, from [30].

0.5 °C was therefore chosen as the minimum tolerance for evaluating the potential of the model-predictive approach. Two supplementary tolerance ranges of ± 1 °C and ± 2 °C were evaluated to show the dependence of performance on the chosen tolerance for different applications. This analysis is carried out for the underlying temperature sensor system and depicted in Figure 5.16.

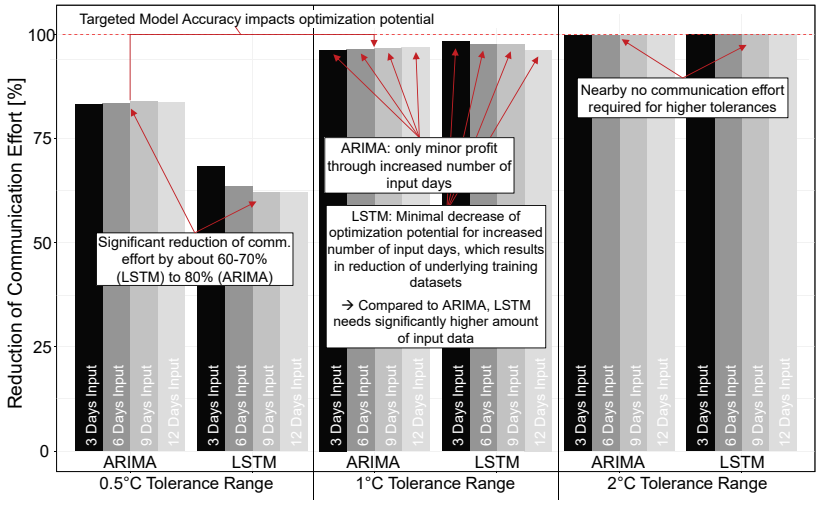


Fig. 5.16: Potential reduction in communication effort provided by using ARIMA and LSTM depending on varying input periods and tolerance ranges. ©[2020] IEEE. Reprinted, with permission, from [30].

A potential reduction of communication effort by more than 60 % with a tolerance of ± 0.5 °C can be observed with both models, with ARIMA reaching more than 80 % potential reduction. LSTM performance decreases with a larger input period, which can be explained by a reduced number of input/output pairs available for training. As expected from the error distribution in Figure 5.15, both models perform almost equally well for higher tolerance ranges with more than a 90 % reduction for ± 1 °C tolerance to nearly no communication effort for ± 2 °C tolerance.

Finally, both modeling approaches have the potential to cut sensor systems' communication effort significantly. Due to its superior overall results and a much higher efficiency in terms of input data needs and computational effort, the ARIMA algorithm is the favored method.

5.1.4.3 Outlook and Future Work

Even requirement profiles of current 5G mMTC applications are in a state of constant evolution. While classic network dimensioning is largely based on stochastic behavior and correlated traffic volumes, the share of event-driven machine communication will increase dramatically. The key challenge will be the realization of reliable critical alarm communication in the face of unpredictable behavior. In this context, a requirement migration towards mission-critical 6G MTC networks is inevitable [429]. To ensure that a guaranteed quality of service can be achieved for application classes that can barely be predicted with a reasonable amount of resources, the available resources must be allocated dynamically and as a function of defined costs (frequency utilization, delay times, energy consumption, …). For this purpose, the definition of new service classes is needed defining targeting latency, Block Error Rates (BLER), and its service characterization, and highlighting the need for future 6G systems to leverage application-domain information about the predictability of resource requirements and conditions. The new service classes are shown in Figure 5.17.

5.1.5 Acknowledgments

In addition to the CRC 876, part of the work has been supported by the federal state of North Rhine-Westphalia and the "European Regional Development Fund" (EFRE) 2014 - 2020 in the course of the "OPUS" project under grant number EFRE0800885, as well as the German Federal Ministry of Education and Research (BMBF) in the project A-DRZ (13N14857).

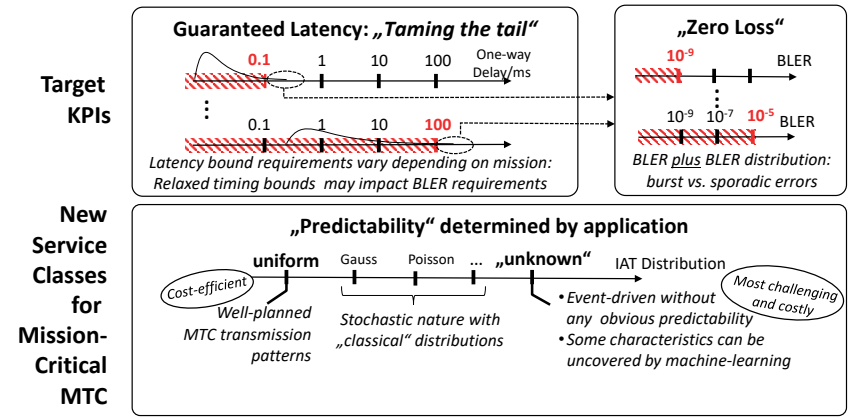


Fig. 5.17: Requirement migration towards mission-critical 6G MTC networks [429]. Used under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

5.2 Resource-Efficient Vehicle-to-Cloud Communications

Benjamin Sliwa

Abstract: Big vehicular data is anticipated to become the new fuel for catalyzing the further development of connected and autonomous driving. Vehicles themselves will act as mobile sensor nodes that actively sense their environment and gather meaningful data for novel crowdsensing-enabled services such as the distributed generation of high-definition maps, traffic monitoring, and predictive maintenance. However, the implied tremendous increase in massive Machine-Type Communication (mMTC) represents an enormous challenge for the coexistence of different resource-consuming applications and entities within the limited radio spectrum. A promising approach for achieving relief through a more resource-efficient usage of existing network resources is the utilization of client-based intelligence. Novel communications paradigms such as anticipatory mobile networking aim to improve decision processes within wireless communication systems by explicitly taking context information into account. In the context of vehicular crowdsensing, these methods exploit the delay-tolerant nature of the targeted applications for scheduling the data transfer with respect to the expected resource efficiency. If the current radio channel and network load conditions do not allow a resource-efficient transmission, the data transfer process is postponed and the acquired data is aggregated locally in favor of a better transmission opportunity in the near future along the expected vehicular trajectory.

In the following, the different evolution phases of the novel Channel-aware Transmission (CAT) scheme are presented. These are characterized by a sequential introduction of different machine learning methods. While the basis CAT approach applies a probabilistic channel-access mechanism based on measurements of the Signal-to-Noise-plus-Interference Ratio (SINR), Machine Learning CAT (ML-CAT) applies supervised learning for predicting the currently achievable data rate using features from the network context, the mobility context, and the application context domain.

This approach is then further extended by Reinforcement Learning CAT (RL-CAT) through the autonomous detection and exploitation of favorable transmission opportunities. Finally, Blackspot-Aware Contextual Bandit (BC-CB) integrates a priori knowledge about the geospatially-dependent uncertainties of the prediction model, which is uncovered by unsupervised machine learning.

It is shown that machine learning-aided opportunistic data transfer is not only able to increase the average data rate of the individual transmissions; it also achieves a massive reduction of the occupied network resources and the power consumption of the mobile device. The price to pay is an increase of the Age of Information (AoI) of the sensor measurements. In addition to the presentation of the novel opportunistic

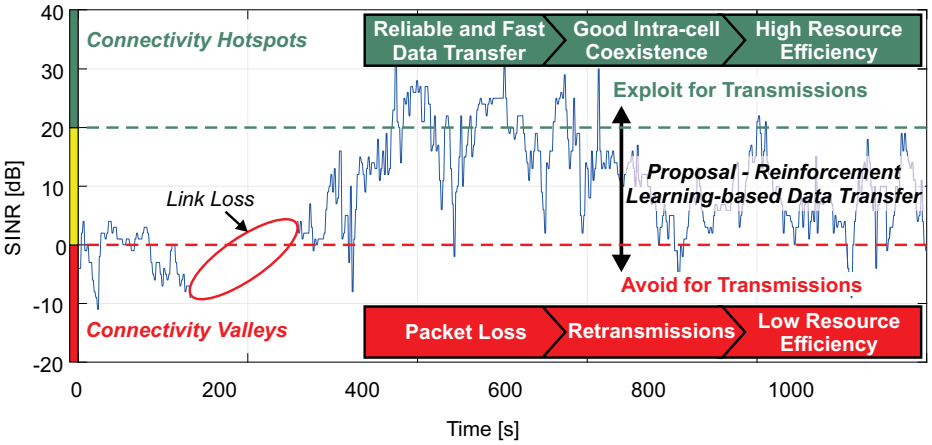


Fig. 5.18: Example for the temporal dynamics of the SINR in vehicular environments.

data-transfer approaches, new machine learning enabled methods for simulating these anticipatory mobile networks are presented, discussed, and validated.

5.2.1 Introduction

According to a recent white paper [6] published by the 5G Automotive Association (5GAA), predictive Quality of Service (QoS)—e.g. the prediction of the data rate along a vehicular trajectory—is expected to become a key enabling method for future connected and autonomous driving.

Although machine learning has already started to penetrate all areas of wireless communications [714], the current 5G standardization efforts focus on implementing intelligence on the network infrastructure side [1]. However, as discussed in initial visionary works [17], it is anticipated that not only the trend of replacing mathematical models with machine learning-based equivalents will continue, but also that pervasive intelligence will be a key driver for the further cellular network evolution. These developments are closely related to the arising anticipatory mobile networking paradigm [107, 630], which aims to improve decision processes within wireless communication systems through explicit consideration of context knowledge and machine learning-based data analysis.

Figure 5.18 shows a real world trace of the Signal-to-Interference-plus-Noise Ratio (SINR) acquired along a vehicular trajectory. It can be seen that vehicular communication channels are characterized by short-term and large-term fluctuations. This behavior is the result of a superposition of distance variations between sender and receiver, mobility-related factors, and obstacle-related signal variation due to shadowing, reflection, and refraction.

In order to guarantee a reliable data transfer, the mobile device reduces the achievable transmission efficiency in favor of better data integrity during challenging radio channel periods. As a result of the implied overhead, a large amount of network resources is unavailable for transmitting the actual payload data.

5.2.2 Related Work

Opportunistic data transfer implements the idea of postponing the data transfer to of delay-tolerant applications to situations where a higher resource efficiency can be achieved due to better radio channel conditions. Acquired data is stored in a local buffer until a favorable transmission opportunity is detected and the whole data buffer is transferred.

Channel-Aware Transmission (CAT) [296, 298], which represents the foundation for the further machine learning-based enhancements presented in this contribution, utilizes Signal-to-Interference-plus-Noise Ratio (SINR) measurements for client-based opportunistic data transmission based on the known significance of downlink quality indicators for assessing the uplink radio channel quality [297]. The probabilistic medium access is performed as

$$p(t) = \begin{cases} 0, & \Delta t \leq \Delta t_{\min} \\ \left(\frac{\text{SINR}}{\text{SINR}_{\max}}\right)^{\alpha}, & \Delta t_{\min} < \Delta t < \Delta t_{\max} \\ 1, & \Delta t \geq \Delta t_{\max} \end{cases} \tag{5.4}$$

whereas Δt is the elapsed time since the last transmission has been performed, $\Delta t_{\min}$ is a minimum inter-packet gap in order to avoid overly frequent medium access, and $\Delta t_{\max}$ is an application-specific deadline for the Age of Information (AoI) of the sensor data packets. Through configuration of α, it can be defined how much the transmission scheme prefers very high metric values within the transmission process.

5.2.3 Machine Learning-Enabled Opportunistic Vehicle-to-Cloud Communication

Although CAT has been demonstrated to achieve significant benefits in comparison with conventional data transmission approaches, recent analyses [638] have shown that physical layer indicators such as SINR have only a limited significance for estimating the achievable data rate. Since the latter is inversely proportional to the transmission duration, it is directly related to the resource occupation time. As a result, the maximization of the end-to-end data rate contributes to improving the intra-cell resource efficiency. For the exploitation of this property, the novel data transfer schemes build upon predictions of the achievable end-to-end data rate.

The methodological evolution of context-aware data transmission approaches is summarized in Figure 5.19. The different evolution stages are characterized through

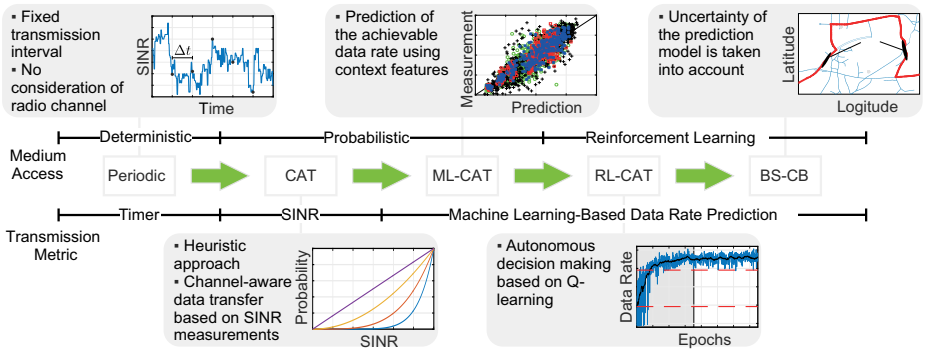


Fig. 5.19: Evolution of channel-sensitive solution approaches for resource-efficient vehicle-to-cloud communication.

a sequential introduction of solution approaches from different machine learning disciplines:

Channel-Aware Transmission (CAT) [296, 298] uses a probabilistic medium-access approach that takes SINR into account.

Machine Learning CAT (ML-CAT) [628, 629, 632] utilizes features from the network, mobility, and application domains for predicting the currently achievable end-to-end data rate, which is then used as the radio channel assessment metric.

Reinforcement Learning CAT (RL-CAT) [636] replaces the heuristic medium access approach with a Q-learning mechanism to autonomously detect and exploit favorable transmission opportunities.

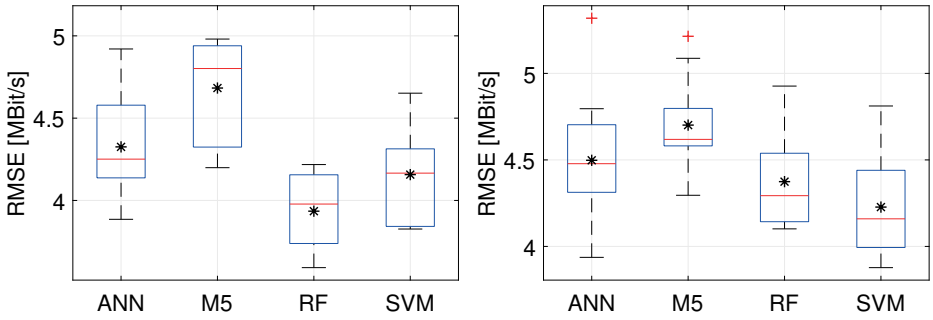
Black Spot-Aware Contextual Bandit (BC-CB) [622, 625] incorporates a priori knowledge about the geospatially dependent uncertainties of the predictions model as a measurement of trust into the latter.

In the following, the enabling methods and novel data-transmission schemes are introduced. Additional details and analyses of various parameter variants are discussed in more detail in the referenced scientific publications.

5.2.3.1 End-to-End Data-Rate Prediction in Vehicular Networks

The considered dataset contains context traces in multiple vehicular evaluation scenarios (campus, urban, suburban, highway). Using the native `Android` Application Programming Interface (API), context indicators from different logical domains are acquired:

- **Network context features $\mathbf{x}_{\text{net}}$**: RSRP, RSRQ, SINR, CQI, TA, Carrier frequency
- **Mobility context features $\mathbf{x}_{\text{mob}}$**: Velocity, Cell Id
- **Application context features $\mathbf{x}_{\text{app}}$**: Payload size of the sensor data packet to be transmitted



(a) Uplink **(b)** Downlink

Fig. 5.20: Comparison of the resulting data-rate prediction accuracy achieved by different machine learning models.

In addition to these passive indicators, the measured data rate of active transmissions in uplink and downlink direction with a random payload size ranging from 0.1 MB to 10 MB is determined every 10 s. An in-depth analysis of the statistical properties of the measurements is given in [638].

Using the resulting feature set $\mathbf{X}$ composed from the individual context vectors $\mathbf{x} = (\mathbf{x}_{\text{net}}, \mathbf{x}_{\text{mob}}, \mathbf{x}_{\text{app}})$, we trained a machine learning model f_{ML} on the corresponding data rate measurements $\mathbf{y}$ such that $f_{\text{ML}} : \mathbf{X} \rightarrow \mathbf{y}$. For this purpose, different regression models are considered:

Artificial Neural Network (ANN) with sigmoid action, two hidden layers, ten neurons per hidden layer, learning rate $\eta = 0.1$, momentum $\alpha = 0.001$, and 500 training epochs.

M5 Regression Tree (M5)

Random Forest (RF) with 100 trees and a maximum tree depth of 15.

Support Vector Machine (SVM) trained via Sequential Minimal Optimization (SMO) with Radial Basis Function (RBF) kernel, regularization parameter $C = 1.0$, and kernel coefficient $\gamma = 1.0$.

The training process is carried out using LIghtweight Machine learning for IoT Systems (LIMITS) [633], which provides high-level automation features for the well-known Waikato Environment for Knowledge Analysis (WEKA) framework and allows the export of `C/C++` implementations of the trained prediction models.

Figure 5.20 shows the Root Mean Square Error (RMSE) of the 10-fold cross-validation in both transmission directions. It can be seen that there are only minor differences for the more complex models if they are properly tuned. Even for the much simpler M5 model, a comparably high prediction accuracy is achieved. The RF model achieves the lowest prediction errors in the uplink direction. In contrast to ANNs and SVMs, another advantage of this approach is a significantly lower complexity for the hyperparameter tuning. As a consequence of these considerations, the further analysis focuses on

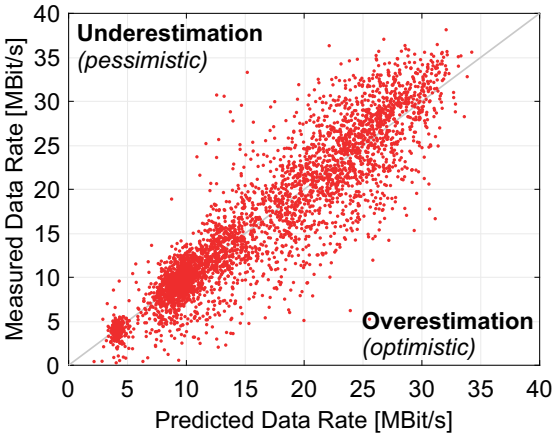


Fig. 5.21: Comparison of RF-based data-rate predictions and corresponding measurements. The diagonal line corresponds to a hypothetical perfect prediction model.

utilizing the RF model for performing the data-rate predictions. A scatterplot of the resulting uplink model is shown in Figure 5.21.

5.2.3.2 Machine Learning CAT (ML-CAT)

The basic idea of ML-CAT is to extend the CAT scheme with a machine learning-based metric for assessing the radio channel quality. While the latter is represented by the predicted data rate $\tilde{S}(t) = f_{\mathrm{ML}}(\mathbf{x}(t))$, the value range of the probabilistic transmission model is implicitly related to the value range of the SINR metric (0 dB to 40 dB according to [298]). Therefore, a normalization $\Theta(t)$ based on the value range $[\Phi_{\min}, \Phi_{\max}]$ of the transmission metric $\Phi(t)$ is defined as

$$\Theta(t) = \frac{\Phi(t) - \Phi_{\min}}{\Phi_{\max} - \Phi_{\min}} \tag{5.5}$$

The transmission probability $p(t)$ is then computed in analogy to Equation 5.4

$$p(t) = \begin{cases} 0, & \Delta t \leq \Delta t_{\min} \\ \Theta(t)^{\alpha}, & \Delta t_{\min} < \Delta t < \Delta t_{\max} \\ 1, & \Delta t \geq \Delta t_{\max} \end{cases} \tag{5.6}$$

5.2.3.3 Reinforcement Learning CAT (RL-CAT)

With RL-CAT, the previously probabilistic medium access is replaced by a reinforcement learning approach. A schematic illustration of the interactions between the different logical entities is shown in Figure 5.22. The model consists of three core components:

– The actual opportunistic data transfer is realized as an **agent** that learns to perform the possible actions—local buffering in expectation of future improvements or

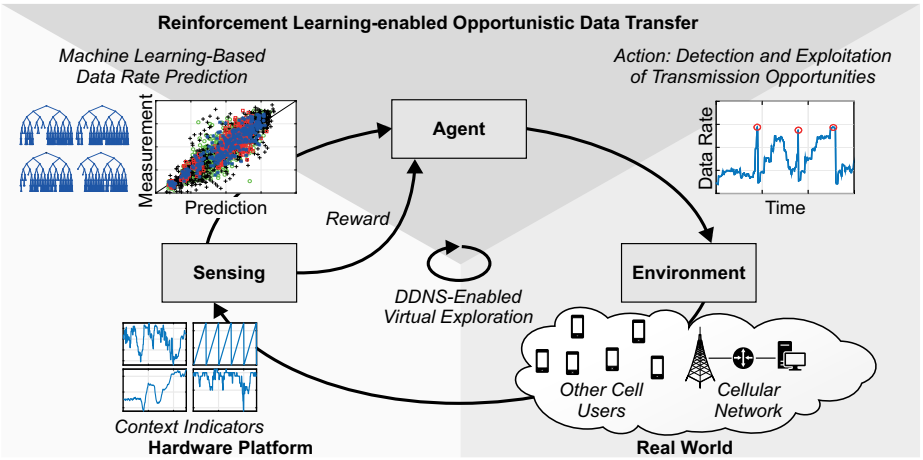


Fig. 5.22: Interaction between the logical entities for reinforcement learning-enabled opportunistic data transfer.

transmission of the whole sensor data buffer—through observation of the resulting *rewards*.

- The **environment** is represented by the cellular network. In contrast to conventional reinforcement learning, which assumes that the actions taken by the agent have a significant impact of the state of the latter within the environment, external impact factors have a dominant influence on the end-to-end behavior.
- **Sensing** is performed using the actual hardware platform based on the measurable context indicators. The raw measurements are brought together using an RF-based data-rate prediction model.

The reinforcement learning-based action selection process utilizes a decision table Q for assessing the expected rewards of the possible actions a_{IDLE} and a_{TX} based on a given state represented by the context tuple $\mathbf{c_t} = \left(\tilde{S}(t), \Delta t\right)$. Based on the available measurements, the action to be executed is determined as $a = \arg\max_a Q(\mathbf{c_t}, a)$. The classical Q-learning update process can be formulated as

$$Q(\mathbf{c_t}, a) = (1 - \alpha) \cdot Q(\mathbf{c_t}, a) + \alpha \left[r_a + \lambda \cdot \arg\max_a Q(\mathbf{c_{t+1}}, a) \right] \tag{5.7}$$

whereas α represents the learning rate, λ is the discount factor, and r_a is the reward of the taken action a. However, as pointed out earlier, the agents impact on its own state can be regarded as negligible: even if the agent was capable of performing "optimal" actions, the achievable end-to-end performance would be still impacted by non-controllable factors such as the network quality and the traffic load caused by other users. Therefore, a myopic approach that focuses on optimizing the immediate reward of the taken actions is implemented by setting $\lambda = 0$, which results in the

simplified formula

$$Q(\mathbf{c_t}, a) = (1 - \alpha) \cdot Q(\mathbf{c_t}, a) + \alpha \cdot r_a. \tag{5.8}$$

The action-specific reward functions are defined as

$$r_{\mathrm{TX}}(S, \Delta t) = \underbrace{w \cdot \frac{(S - S^*)}{S_{\max}}}_{\text{Data rate optimization}} + \underbrace{(1 - w) \cdot \frac{\Delta t}{\Delta t_{\max}}}_{\text{AoI optimization}} \tag{5.9}$$

and

$$r_{\mathrm{IDLE}}(\Delta t) = \begin{cases} \Omega & \Delta t \geq \Delta t_{\max} \\ 0 & \text{else} \end{cases}. \tag{5.10}$$

Hereby, the parameter w allows us to control the fundamental trade-off between data-rate optimization and AoI reduction, S^* is the target data rate, and $S_{\max}$ represents the upper data rate bound of the empirical measurements. Although there is no immediate reward if no data transfer is initiated, Ω serves as a punishment factor if the buffering time Δt exceeds the application-specific deadline $\Delta t_{\max}$.

Instead of performing a large number of real-world transmissions for training the reinforcement learning mechanisms, a Data-Driven Network Simulation (DDNS) setup is implemented according to [637]. In contrast to classical system-level network simulation, which requires a large number of assumptions and simplifications for setting up virtual representations of concrete real world scenarios, DDNS makes use of a combination of machine learning models and empirical context traces. This black spot approach does not require us to explicitly model communicating entities and achieves not only a close-to-reality representation of real-world behavior but also a massive computational efficiency.

5.2.3.4 Black Spot-Aware Contextual Bandit (BS-CB)

While BS-CB builds upon the reinforcement learning-based medium access approach of RL-CAT, it introduces additional mechanisms for accessing trust in the data-rate predictions. Moreover, it replaces the Q-learning component by a contextual bandit reinforcement model. A detailed description of BC-CB is given in [622].

In order to improve the data-rate prediction accuracy, the concept of black spot regions is introduced. Within those areas, the properties of the geographical environment lead to a significant increase in the location-specific prediction error (e.g., related to an increased handover probability). If knowledge about the presence about those black spots is available, transmissions can be postponed in order to avoid severe mismatches of predictions and measurements. For this purpose, BS-CB leverages a priori data about previous measurements in the targeted scenarios. Based on k-means-enabled unsupervised learning, the data is clustered into N_c clusters. For each cluster, the clusterwise RMSE is computed and compared with a given cluster threshold $\mathrm{RMSE}_{\max}$. All clusters that exceed the defined threshold are treated as black spot clusters and fitted to ellipses.

During the application phase of the transmission mechanism, the vehicle performs an ellipse test to check if it is currently within a black spots region. If the condition is fulfilled, the transmission process is postponed.

The actual reinforcement learning process is modeled as a contextual bandit that proposed an action a (either `IDLE` for further local buffering or `TX` for data transmission) using

$$a = \arg\max_{a \in \mathbf{A}} \left(\underbrace{\hat{\theta}_a^T \mathbf{c}}_{\text{Estimated reward}} + \underbrace{\alpha \sqrt{\mathbf{c}^T \mathbf{A}_a^{-1} \mathbf{c}}}_{\text{CB}} \right) \tag{5.11}$$

whereas $\hat{\theta}$ corresponds to the ridge regression coefficients of action a. $\mathbf{c} = \left(\tilde{S}(t), \Delta t \right)$ is the d-dimensional context tuple for the predicted data rate $\tilde{S}(t)$ and the current buffering delay Δt.
The degree of exploration is controlled using the greediness parameter δ

$$\alpha = 1 + \sqrt{\frac{\ln(2/\delta))}{2}}. \tag{5.12}$$

After either the `IDLE` or the `TX` has been performed, the regression coefficients are updated as

$$\hat{\theta}_a \leftarrow \mathbf{A}_a^{-1} \mathbf{b}_a \tag{5.13}$$

with

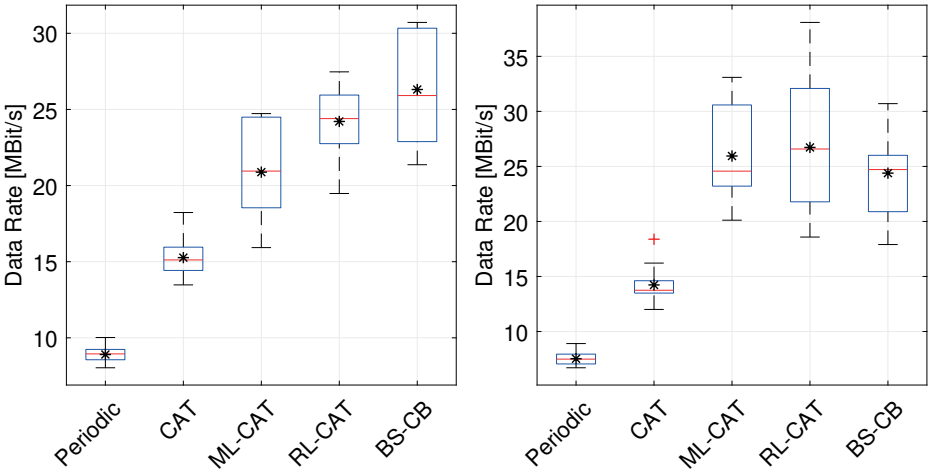
$$\mathbf{b}_a \leftarrow \mathbf{b}_a + r_a \cdot \mathbf{c}. \tag{5.14}$$

For determining the actual rewards of the chosen actions, the reward functions of RL-CAT are re-utilized according to Equation 5.9 and Equation 5.10.

5.2.4 Results of the Real-World Performance Comparison

For the performance evaluation of the novel machine learning-enabled methods, a 25 km long evaluation track with varying environmental characteristics, speed limitations, and building densities is considered. For each transmission scheme, ten real-world drive tests are performed. Hereby, a virtual sensor application generates 50 kB of sensor data per second, which is buffered locally until a transmission decision is taken for the whole buffer.

Figure 5.23 shows the achieved end-to-end data rate of the transmission schemes. While the basic channel-sensitive approach of CAT is already able to achieve a significant improvement of the data rate, the latter is highly increased through the introduction of machine learning-based channel assessment. Moreover, the reinforcement learning-based data transfer results in additional gains. In comparison to conventional fixed interval data transmission, BS-CB achieves performance improvements of 195 % in uplink and 223 % in downlink direction.



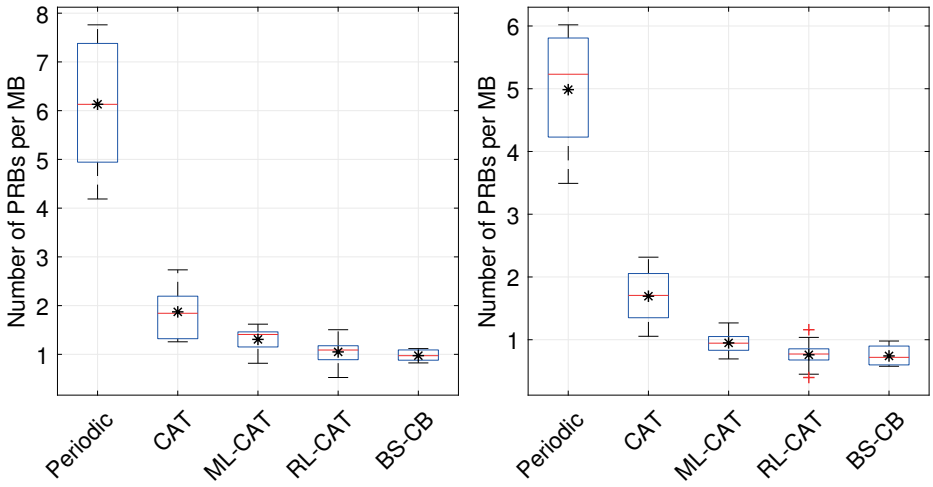
(a) Uplink **(b)** Downlink

Fig. 5.23: End-to-end data rate of the different transmission schemes.

As shown in Figure 5.24, the apparently selfish goal of data rate maximization contributes to improving the good of all: all opportunistic data transfer methods are able to achieve a significantly better resource efficiency than the conventional approach. Although the methodological evolution is also represented in the achieved results, there are only minor differences between the machine learning-enabled methods. Here, BS-CB reduces the number of occupied cell resources by around 85 % in both transmission directions.

5.2.5 Outlook and Future Work

Due to its enabling character for all presented transmission schemes, future work should focus on optimizing the accuracy of the data-rate prediction model. A major limitation of client-based prediction approaches is their limited insight into the current traffic load within the cell. Future networks could compensate this limitation through active announcement of network infrastructure knowledge about the traffic load, e.g., acquired through novel 5G mechanisms such as the Network Data Analytic Function (NWDAF). As shown by a recent feasibility study [626], the integration of available network knowledge reduces the RMSE by 25 % in the uplink and 30 % in the downlink direction.



(a) Uplink **(b)** Downlink

Fig. 5.24: Resource efficiency of the different transmission schemes.

5.3 Mobile-Data Network Analytics Highly Reliable Networks

Robert Falkenberg
Karsten Heimann
Benjamin Sliwa

Abstract: The analysis of mobile network data is a fundamental requirement for the development and invention of novel networking approaches that fulfill the rapidly growing requirements and demands on those networks. This process requires the identification and a thorough investigation of shortcomings in existing field deployments, independent of the network operators and/or the network equipment vendors. Despite public standardization by the 3rd Generation Partnership Project (3GPP), cellular networks are developed and operated as closed systems that provide a predefined networking service to the subscriber while disclosing only a minimum of system-related information such as signal strength or quality in the User Equipment (UE).

However, researchers often require a deeper insight into network functionality, especially when it comes to considering network load and occasional congestions while still maintaining the privacy of the regular network users. With this knowledge, future devices may predict their achievable throughput passively under the current load and channel conditions without the need of triggering a transmission just for the sake of throughput measurements that in turn induces (unnecessary) network load. They may leverage this predicted information for e.g. load balancing, network selection, or service adaptation. Since cellular networks are centrally governed by the base stations, which assign the spectral resources by explicit and fine-grained signaling to each active device in the coverage area, information about the cell-wide resource utilization is already "in the air". For performance reasons and unlike the ciphered payload exchange between UE and the base station, the control messages that carry the resource assignments are not encrypted.

However, these messages are scrambled by a device- and session-specific Radio Network Temporary Identifier (RNTI), which is essential for the proper interpretation and validation of those messages and which is exchanged only once at the beginning of each session. This section describes the achievements of the CRC 876 in extracting these control messages of new and already active sessions efficiently and reliably over the air and without the need for expensive specialized hardware. The methodology of the underlying control channel analysis is embedded into a comprehensive open-source software framework Fast Analysis of LTE Control Channels (FALCON), which uses Software-Defined Radios (SDRs) to capture the base station's signal and accurately extracts the control messages in real time on a regular computer system.

In subsequent case studies, supervised learning is used to leverage the disclosed network-load information from short-term observations for the prediction of the expected data rate and ultimately uses this as a metric for dynamic network selection to achieve the highest throughput over the fastest network connection currently available.

5.3.1 Introduction

The steady increase in data traffic in mobile networks, triggered by the rapidly growing number of human and machine network subscribers, poses a challenge to both network operators and the services that depend on them in face of the limited radio spectrum for meeting the simultaneously growing demands on quality of service. Achievable data rates depend on the one hand on the cell bandwidth and signal quality, and on the other hand on the activity of other cell users competing simultaneously for the available radio resources. One of the possible strategies is to use higher frequency ranges, in which higher bandwidths and thus more spectral resources are available for transmission. However, due to the inherently higher signal attenuation, these frequency ranges are only suitable for covering smaller areas, so that region-wide coverage is only economical with a correspondingly high user density. In order to meet the growing demands in the remaining areas and to counteract bottlenecks, the usage efficiency of the available resources must therefore be further increased. For example, subscribers could switch to less busy networks or perform delay-tolerant data transmissions only when channel and load situations are favorable.

However, both the research and the application of such mitigation strategies require the accurate measurement of both signal quality and instantaneous network load in order to identify overload situations without creating unnecessary load themselves, e.g. in the form of test transmissions. Even though mobile devices measure signal strength and quality autonomously, present mobile networks allow users and external observers only a very limited insight into the momentary resource utilization of the cell. Although the total occupancy of radio resources can be determined by spectrum analysis (cf. top row in Figure 5.25), the actual degree of contention in the case of full occupancy remains concealed since the number of served subscribers cannot be identified in the spectrum (cf. last two columns in Figure 5.25).

In 4G and 5G networks, the distribution of spectral resources is governed by the base station, which explicitly allocates its resources to single active subscribers via special control channels. For efficiency reasons, these allocation messages are not encrypted, but reliable decoding requires knowledge of the addressee's RNTI, since the attached checksum is scrambled with it. As a result, UEs can only read the assignments that affect them. Inactive users without assigned RNTI and external observers can decode only the assignments for specially reserved RNTIs that concern general system information or paging.

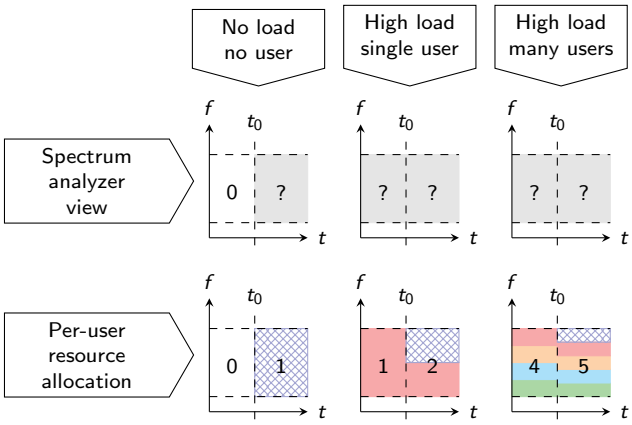


Fig. 5.25: If a new user starts a transmission at t_0, the number of allocated spectral resources (shaded in blue) depends on the number and activity of other participants (bottom row, other colors). In the spectrum analysis (upper row), the allocations of individual users are indistinguishable, and the number of participants remains unknown. ©[2016] IEEE. Reprinted, with permission, from [195].

In the course of the CRC 876 research, efficient control channel analysis methods for finding valid RNTIs have been developed and evaluated, enabling passive load sensing of the mobile network and thus providing valuable information for a client-side data-rate prediction. The presented approaches are directly applicable to public 4G cellular networks and enable real-time discovery of all resource allocations using off-the-shelf PCs and SDRs. A comprehensive reference implementation is provided in the form of the open-source framework FALCON. Using the collected data and derived features to characterize the network load, supervised learning is used to train prediction models that enable data-rate prediction whose accuracy significantly exceeds previous approaches based purely on signal strength. Applied simultaneously to multiple cellular networks, the prediction enables the UE to perform predictive network selection in order to transmit data over the network with the most promising data rate, especially during high-load periods. The prediction accuracy, achievable data rate gain, and impact on UE energy consumption are evaluated using case study data collected in public mobile networks.

The following sections are structured as follows: Section 5.3.2 presents related work in the area of control channel analysis. Subsequently, Section 5.3.3 discusses methods for analyzing control channels and presents some further implications that can be derived from observing the cell activity. Key findings are summarized and a conclusion is drawn in Section 5.3.5.

5.3.2 Related Work

The assessment of the current mobile network connection, especially in terms of link throughput, is commonly done by means of active probing [289]. This means that a data transmission is triggered to measure the throughput currently available for the device under test. However, this approach loads network and radio resources, which may be omitted by a purely passive measurement or prediction.

The device itself has only a limited view on the current network load, yet it provides performance indicators such as Reference Signal Received Power (RSRP) and Reference Signal Received Quality (RSRQ), which can be used only for a rough forecast of the achievable data rate of subsequent transmissions [297]. Authors in [392] additionally utilize details from lower protocol layers and the chipset.

More promising solution approaches need to consider further information that is usually outside the scope of the mobile device. For this reason, expensive commercial tools with special hardware requirements[695] as well as off-the-shelf SDRs and open-source protocol stacks allow tailored solutions based on deep insights into the signaling protocol behavior and related routines. In terms of the SDR-based approach, especially LTEye from [359] and Online Watcher for LTE (OWL) from [108] deal with the analysis of the control channel for resource allocations to infer the current resource utilization and concurrently active users. As will be detailed in the next section, LTEye suffers from numerous false-positive detections, while OWL constitutes a solid, real-time capable approach which only detects new devices though. In contrast to that, our approach FALCON [196] implements improved detection capabilities of the resource utilization in mobile networks and is even able to forecast or recommend the most performant network at a given time.

5.3.3 Control Channel Analysis

In current mobile networks, radio resources are divided by time and frequency in the manner of a two-dimensional resource grid. The resource grid spans the cell bandwidth in frequency domain and the time domain is divided into a nested and periodic structure of symbols, slots, subframes, and frames. The smallest resource unit in 4G and 5G networks is the Resource Element (RE), which corresponds to a single subcarrier of an Orthogonal Frequency Division Multiplexing (OFDM) symbol. According to a predefined pattern, some REs are used to broadcast synchronization sequences or provide reference levels for equalization. REs without a special purpose serve as resources for the transmission of any other data, including control and payload messages. These spare REs are grouped into equal-sized Resource Blocks (RBs), which are the smallest unit of resources that can be allocated to individual UEs. In common 4G Networks, a RE spans 12 subcarriers in frequency domain and 7 symbols in time, which corresponds to

a bandwidth of 180 kHz and a duration of 0.5 ms. However, resource allocations always apply for both slots (0.5 ms) of a subframe (1 ms).

The allocation of RBs is organized centrally by the base station and signaled to the UEs via a special control channel, namely Physical Downlink Control Channel (PDCCH), which is located in the first 1, 2, or 3 symbols of every subframe and spans the entire cell bandwidth. It includes assignments for transmissions in both directions, the downlink (i.e. from the base station to the UE) and the uplink (i.e. from the UE to the base station). These apply in the downlink (DL) direction for the current subframe or in the uplink (UL) direction 4 subframes later to give the UE enough time to prepare.

From a logical point of view, the PDCCH consists of a sequence of Control Channel Elements (CCEs), each comprising 36 REs, whose total number is calculated from the cell bandwidth and the number of occupied OFDM symbols. These CCEs carry the encoded Downlink Control Information (DCI) for single UEs, which contain the RB allocation, the Modulation and Coding Scheme (MCS), the power control commands, and further control information required for decoding or encoding the payload in the allocated resources. Base stations use rate 1/3 channel coding, interleaving, and rate matching for each emitted DCI data structure to provide FEC and to fit the encoded sequence into L, $L \in \{1, 2, 4, 8\}$ consecutive CCEs. The aggregation level L is selected by the base station according to the channel conditions of the addressee to ensure proper reception. Any additional or spare space within the L CCEs is filled by cyclic repetitions of the encoded sequence and interleaving ensures an even distribution of repeated bits.

Prior to the encoding, each DCI is appended with its 16-bit Cyclic Redundancy Check (CRC) checksum, which is additionally scrambled (via binary XOR) with the RNTI of the addressee. Conversely, receiving UEs only consider decoded DCI where the CRC matches their current RNTI.[1]

Since the PDCCH has no table of contents, only blind decoding of the CCEs can determine whether relevant information is present. This also includes all possible combinations resulting from different L. To reduce the number of decoding attempts for a UE, the standard defines a search space function that restricts the search space to a maximum of 22 evenly distributed locations according to RNTI, subframe, and L.

Furthermore, the standard defines numerous DCI formats for different transmission modes, which depend on the number of antennas used and the capabilities of the UE and the evolved NodeB (eNodeB) as 4G base station. Transmission modes are negotiated both when connections are established and dynamically depending on the channel conditions. The DCI formats differ in their length and consequently in the length of the encoded sequence. However, the same circular approach is used for rate matching, i.e. to populate the CCEs so that the initial format is no longer apparent in the transmitted

1 The UE also tracks special reserved RNTIs for system information and paging as required.

sequence. End devices therefore decode the sequences multiple times, assuming any DCI formats (or their lengths) that comply with the specified transmission mode.

Consequently, an external observer faces the following challenges when decoding the entire content of the PDCCH:

1. Numerous decoding attempts covering all locations of the PDCCH including all possible combinations of DCI format length and aggregation level are computationally expensive and result in many invalid DCI candidates.
2. Discard all DCI candidates with CRC that do not match a valid RNTI.
3. Find valid RNTIs within the set of decoded DCI candidates.

The chicken-or-egg situation resulting from the last two points can be resolved in several ways: LTEye [359] re-encodes each decoded DCI candidate, compares the encoded sequence with the received bits on the channel, and discards any candidates that deviate by a certain degree. But in the presence of noise or interference, we show that this approach is highly inaccurate and leads to numerous false decisions. In a more robust approach, [108] follows the initial connection establishment of joining UEs, which contains the RNTI assignment in plain text, and builds up a list of valid RNTIs. However, RNTIs of UEs that entered the cell before the monitoring remain undetected. Therefore, OWL follows the approach of [359] as a fallback. By contrast, the authors of UnCover Information in Mobile Access Networks (U-CIMAN) [771] propose to first accept any DCI candidate and to decode the potential payload in the allocated RBs. If this attempt fails due to an invalid CRC of the payload, the DCI is discarded. The approach involves a significant computational cost due to the larger amount of data and the more complex decoding procedure for the payload data.

In this area, CRC 876 has made substantial contributions to a resource-efficient yet reliable control channel analysis that is especially suitable for short-term monitoring in order to estimate the total cell load. In [195] we propose a histogram-based approach in conjunction with an inverse application of the search space function to identify valid RNTIs and decode the corresponding DCI candidates. First, DCI candidates decoded from all possible locations, formats, and aggregation levels are validated with respect to their permitted positions, as the eNodeB never places DCI outside their associated search space. This approach reliably filters 80-90 % of all candidates, including invalid DCI.

The following filter stage first collects the RNTIs of all DCI candidates in a history. According to an attached histogram, all candidates are discarded whose RNTI in the histogram does not exceed a threshold value k. It is based on the fact that active participants receive multiple assignments within a short period of time and that their RNTIs occur more frequently than the random RNTIs that result from decoding with incorrect parameters. An example is given in Figure 5.26.

Length of history and threshold value are optimization parameters that allow a trade-off between the probability of false positive detection, the minimum required activity of individual UEs, and the detection delay [194].

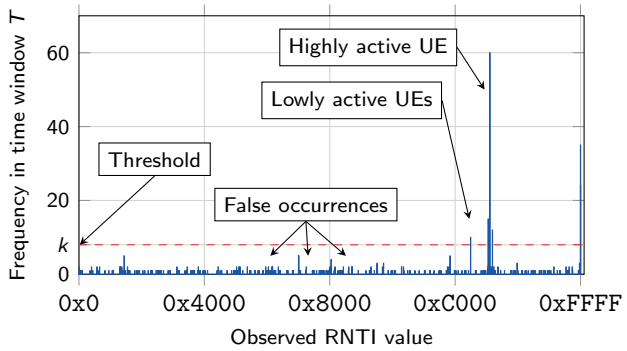


Fig. 5.26: Example for the histogram-based RNTI validation approach. RNTIs of active UEs appear with high frequency, while RNTIs of false DCI uniformly spread over the entire value range with low frequency. ©[2016] IEEE. Reprinted, with permission, from [195].

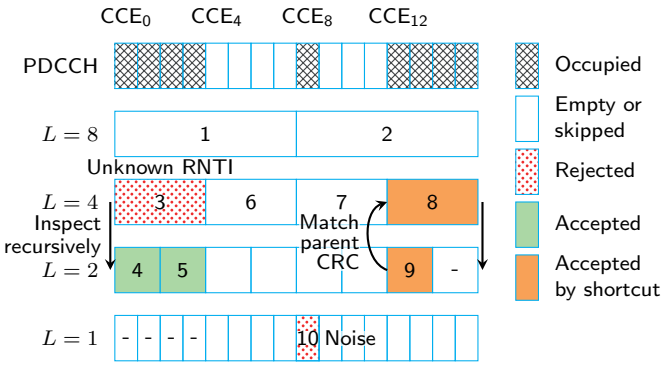


Fig. 5.27: Recursive PDCCH analysis with the *short-cut* decoding approach for immediate yet reliable discovery of unseen RNTIs. ©[2019] IEEE. Reprinted, with permission, from [196].

To improve the detection speed of unseen RNTIs, which is especially important for short-term observations, we proposed a novel *short-cut* decoding approach in [196]. The approach exploits the scheme of how the eNodeB populates the CCEs with the encoded DCI sequence. Although in most cases such sequence fits into a single CCE, operators configure the eNodeB to use higher aggregation levels in order to increase the robustness against distortions. Conversely, due to the circular repetition, a properly cropped sequence still allows for a correct decoding of the DCI. Therefore, if both decoding of the full and the shortened sequence result in the same DCI and CRC, the associated RNTI can be assumed as valid and the DCI shall be accepted. This approach can be implemented efficiently by combining a breadth-first search with a depth-first search for each location as shown in Figure 5.27. The top line shows the PDCCH as a sequence of consecutive CCEs, which are either occupied or empty according to the placement done by the eNodeB. Empty CCEs, recognizable by insufficient signal

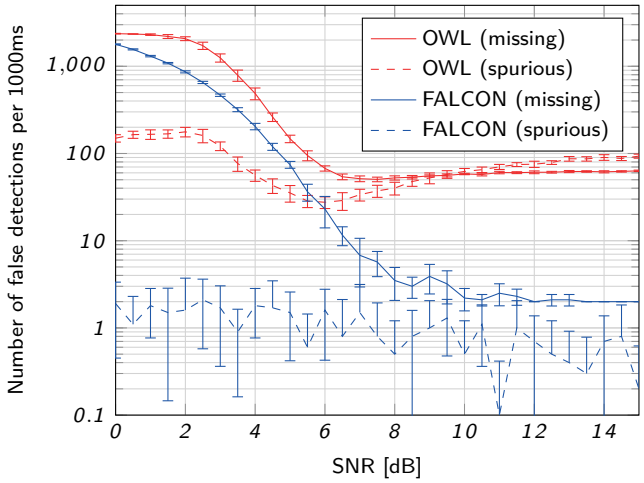


Fig. 5.28: Simulation results for decoding the PDCCH of 1 s Long Term Evolution (LTE) signal with FALCON and OWL for different Signal to Noise Ratio (SNR) values. Results are averaged over 10 repetitions; the standard deviation is shown as error bars.

power, can be skipped as they cannot contain any meaningful information. Conversely, occupied CCEs may also not have meaningful information, as the received signal power may originate from a neighbor cell with overlapping PDCCH in time and frequency. The breadth-first search component starts with aggregation level $L = 8$ and sequentially decodes all locations at this level (second line). In the given example, the two possible locations (1) and (2) are skipped, as each overlaps at least one empty CCE, and hence does not form a continuous sequence. The search continues with $L = 4$, inspecting the first location (3) with continuous CCE occupation by decoding the sequence for all DCI formats. If this inspection does not result in any DCI with a known RNTI, the depth-first component is activated and the location is inspected recursively using the next-smaller aggregation level. In the given example both locations (4) and (5) contain DCI with known RNTIs; overlapping locations (e.g. for $L = 1$) are marked as checked. As the recursion terminates, the breadth-first search continues with locations (6) and (7) both being skipped. Next, location (8) contains a valid DCI at $L = 4$ but the RNTI has not yet been seen. However, the recursive inspection of the shortened sequence, given by the first half at location (9), returns the same DCI and RNTI. As this only happens for valid DCI, the RNTI is immediately added to the list of known RNTIs, the DCI is accepted, and overlapping locations are marked as checked.

To enable detection even in the case of poor signal quality, where the bisected sequence can no longer be decoded correctly, histogram-based validation can be employed afterwards. If a recursive descent does not discover a known RNTI, all potential RNTIs along the descent path are added to the history and RNTIs exceeding a threshold are added to the active set as described above.

Robustness and reliability of the combined approach in FALCON is presented in Figure 5.28. It shows the number of missed and false DCI messages as solid and dashed lines, respectively, as a function of the SNR on the channel after analyzing a well-defined LTE signal for the duration of 1 s. Starting with 0 dB for poor radio conditions, the SNR is increased in steps of 0.5 dB to 15 dB, representing an excellent signal. For each step, the figure shows the average and the standard deviation over 10 repetitions. Furthermore, the figure also contains the results of OWL, which relies on the re-encoding approach for short-term observations.

Independently of the SNR, the amount of spurious DCI messages stays at a negligible level at FALCON, whereas OWL produces at least 30 and up to 100 false detections. In general, for all covered SNR, FALCON misses significantly fewer DCI messages than OWL. Especially for SNR values greater than 7 dB the number of missed DCI messages undershoots 10 for FALCON, while OWL remains on a level between 50 and 100. Thus, the comparison of both approaches reveals the robustness and reliability of FALCON.

Similar results are achieved in the field as an activity histogram of each RNTI over 5 s as shown in Figure 5.29. Blue circles represent the number of resource allocations detected by FALCON for each RNTI, and red crosses show the results of OWL. It is evident that the most active RNTI concentrate in a small value interval, indicating that the eNodeB assigns RNTIs consecutively to new UEs. The peak region moves over time towards larger RNTI values, as shown in the results 5 min and 10 min later. On the other hand, OWL reports numerous spurious DCI messages with random RNTIs, which are uniformly distributed over the entire value range with very low frequency.

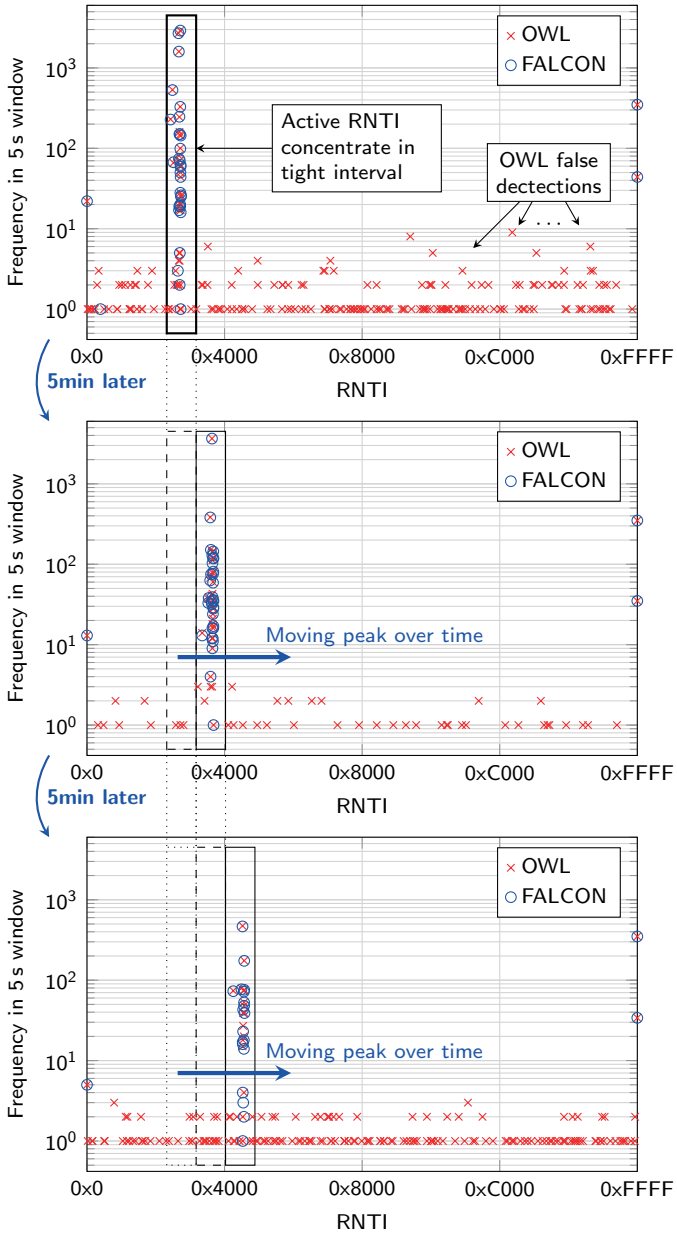


Fig. 5.29: Activity histogram of individual RNTI detections from field measurements. The highlighted peak moving over time indicates a RNTI assignment strategy of the base station. ©[2019] IEEE. Reprinted, with permission, from [196].

5.3.4 Cooperative Data Rate Prediction Leveraging FALCON

Client-based data-rate prediction is a key enabler for anticipatory mobile networking. By considering network context measurements to estimate the end-to-end transmission efficiency—represented by the predicted data rate—mobile clients can actively contribute to optimizing the intra-cell resource efficiency by scheduling data-intense transmission to resource-efficient connectivity hotspots [622]. However, the accuracy of client-based data-rate prediction methods is inherently limited since the UEs are only aware of the radio channel conditions but not of the network load.

In addition to the passive context measurements of purely client-based data-rate prediction according to [622], FALCON allows the derivation of additional features (number of Physical Resource Blocks (PRBs) and UEs, Transport Block Size (TBS), MCS) that are correlated to the current network load of the cell. For a proof-of-concept evaluation, the following feature sets are derived for the two transmission directions after an initial feature importance analysis:

- **Uplink feature set**: RSRP, RSRQ, velocity, payload size, number of PRBs, number of UEs, cell ID
- **Downlink feature set**: RSRQ, velocity, payload size, number of PRBs, number of UEs, TBS, MCS, cell ID

It can be seen that the uplink direction is more sensitive to the radio channel conditions while the downlink performance highly depends on the intra-cell traffic load.

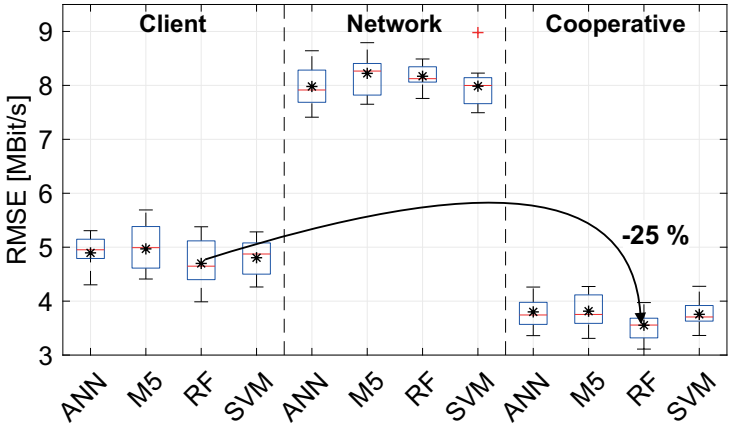


Fig. 5.30: End-to-end uplink data-rate prediction: performance comparison of different prediction approaches and machine learning models. ©[2020] IEEE. Reprinted, with permission, from [626].

The resulting prediction accuracy of different machine learning models (Artificial Neural Network (ANN), M5 regression tree, Random Forest (RF), Support Vector Ma-

chine (SVM)) is shown in Figure 5.30. Although the RF model achieves the highest overall accuracy, there are only minor differences between the machine learning models. However, significant differences can be observed for the three different data-rate prediction methods. While the purely client-based approach is mostly unaware of the traffic load and the purely network-based approach yields a high prediction error due to the absence of radio channel information, the cooperative prediction method reduces the average Root Mean Squared Error (RMSE) by 25 % in the uplink direction. As further analyzed in [626], similar improvements are also achieved in the downlink direction.

These initial results show that the context-awareness and the predictability of mobile communications can be significantly improved by combining client measurements with network-side information. Therefore, future networks such as 6G should actively provide network-load information to the clients in order to allow them to actively participate in network management functions.

5.3.5 Conclusion

FALCON is a novel open-source and SDR-based instrument for LTE control channel analysis that allows the reliable monitoring of the resource allocations of LTE cells in real-time. Through the application of shortcut-precoding, a fast DCI integrity check is achieved and the list of active RNTIs—which corresponds to an estimation of the number of active users—is derived. Aided by a histogram approach, the accuracy of FALCON is maintained even during low signal quality periods. The revealed network-side information is of particular value for intelligent networking methods that utilize end-to-end predictions for their decision making, such as resource-efficient vehicle-to-cloud communications that is discussed in Section 5.2. As purely client-based data-rate prediction approaches that rely on network context measurements are unaware of the current network load, their achievable prediction accuracy is inherently limited. As demonstrated in a first real-world proof-of-concept study [626], the incorporation of FALCON offers the potential to improve client-based data-rate prediction methods by up to 25 % in the uplink and 30 % in the downlink direction.

5.4 Machine Learning-Enabled 5G Network Slicing

Caner Bektas
Fabian Kurtz
Dennis Overbeck
Christian Wietfeld

Abstract: In this contribution, the different parts of the end-to-end network slicing concept are presented, including the Core Network (CN) and the Radio Access Network (RAN), while highlighting the differences and similarities of both domains.

Further, prototypical implementations and empirical evaluations of 5G network slicing are discussed, deepening the understanding of network slicing and identifying possible advantages and challenges. The predictability of user traffic in the respective network slices poses such a challenge, as resources in the RAN–in contrast to resources in the CN–are subject to fluctuations based on channel quality. Critical infrastructures typically require very low latencies in the single-digit milliseconds range and are thus considered ultra-Reliable Low Latency Communication (uRLLC) . To mitigate latency-intensive scheduling requests and grant operations, resources in the RAN have to be pre-allocated for uRLLC slices.

This operation, also known under the term Configured Grants (CGs), pre-allocates resources for, say, high-priority slices, so that User Equipments (UEs) are able to send data without asking for resources, which reduces the scheduling latency down to zero. The simplest method for calculating CGs is based on static allocations, which has one major drawback: unused resources are wasted, and thus, can not be used by remaining slices, effectively lowering spectral efficiency. Here, we present SAMUS (Slice-Aware Machine Learning-based Ultra-Reliable Scheduling), a data-driven method to predict resources in the future based on real data, e.g., solar activity in smart grid slices, to reduce latencies while maintaining high spectral efficiency.

5.4.1 Introduction to 5G End-to-End Network Slicing

Critical infrastructures, such as energy networks, logistics, or autonomous transportation, are becoming more and more automated to further increase efficiency. Automation is often achieved by the self-organization of processes and actors via mobile communication systems. As many different vertical industries are reliant on mobile communication, a highly diverse set of Key Performance Indicators (KPIs) need considering by the communication systems.

Until recently, dedicated communication networks were the go-to solution in order to meet these divergent criteria, as they can be designed specifically for the needs of the respective critical infrastructures. Consequentially, the fifth generation of mobile networks (5G) aims to unify these different and partly contradictory set of requirements into a single physical infrastructure. Employing Software-Defined Networking (SDN) and Network Function Virtualization (NFV) techniques, 5G *network slicing* is integrated into the 5G standard. By utilizing virtual dedicated networks called network slices on top of a single physical communication network, various vertical industries can be automated, as shown in Figure 5.31.

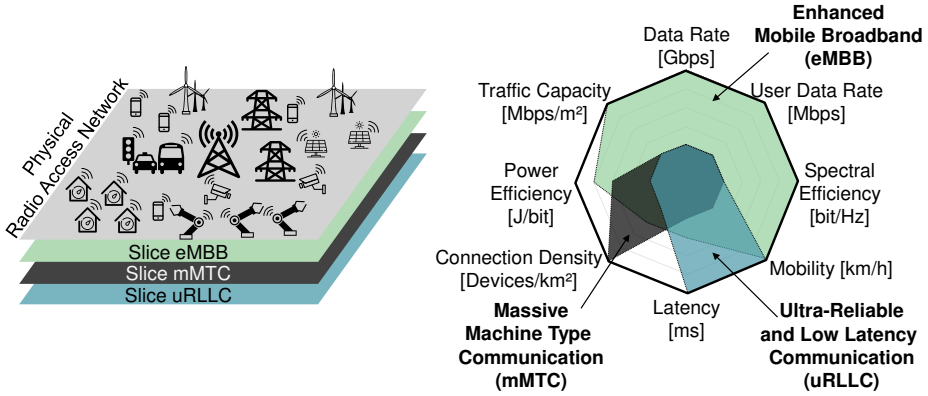


Fig. 5.31: Network slicing as a key enabler for fulfilling all specific service requirements simultaneously.

5.4.2 5G Core Network Slicing

5.4.2.1 Description and Methodology

The virtualization of network resources depicts a main pillar of 5G networking as illustrated in Figure 5.32.

The creation of multiple isolated network partitions known as slices can independently and efficiently manage different use cases with their respective demands on QoS or other guarantees. For 5G communication, three main categories are defined. The first category is enhanced Mobile Broadband (eMBB), which is used for data-rate intensive services (up to 20 Gbit/s). This category comprises ultra-high resolution video streaming as well as fixed wireless broadband and Augmented respectively Virtual Reality (AR/VR) . The second category is massive Machine Type Communication (mMTC) , designed for the emerging Internet of Things (IoT) and Industry 4.0 applications, both of which introduce a significant increase in inter-machine communication. But

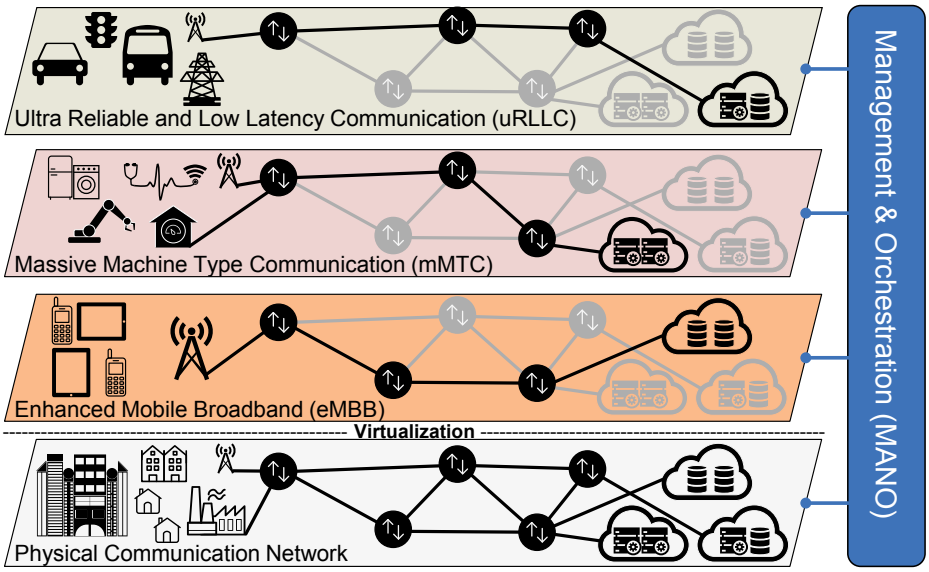


Fig. 5.32: Overview of a sliced 5G communication network.

a massive amount of devices comes with its own set of challenges in, say, electrical power grid application scenarios such as smart metering. The third major category is the ultra-Reliable and Low Latency Communication (uRLLC) service. This category comprises services such as Intelligent Transportation Systems (ITS) with Floating Car Data (FCD)-based Vehicle-to-X communication. Here, mission-critical and latency-sensitive applications are addressed. Our approach builds on NFV and SDN, which are closely related. With NFV, hard- and software is decoupled and functionalities are abstracted in order to achieve highly flexible communication infrastructures for enabling cloud computing. Virtual Network Functions (VNFs) are now able to run on Commercial-Off-The-Shelf (COTS) server platforms . By using the complementary approach of NFV and SDN, the controller can dynamically route traffic flows between the VNFs, while being deployed as a VNF itself. In addition to the utilization of SDN and NFV, our concept is based on queuing strategies utilizing the Hierarchical Token Bucket (HTB) . On the bare-metal and virtualized data-plane devices, the switching software Open vSwitch (OVS) is deployed. Furthermore, a Management and Network Orchestration (MANO) controller is implemented. This controller creates a main bridge in each switch, which includes the respective physical ports. By that, one bridge per slice is added or removed as needed. The concept is depicted in Figure 5.33.

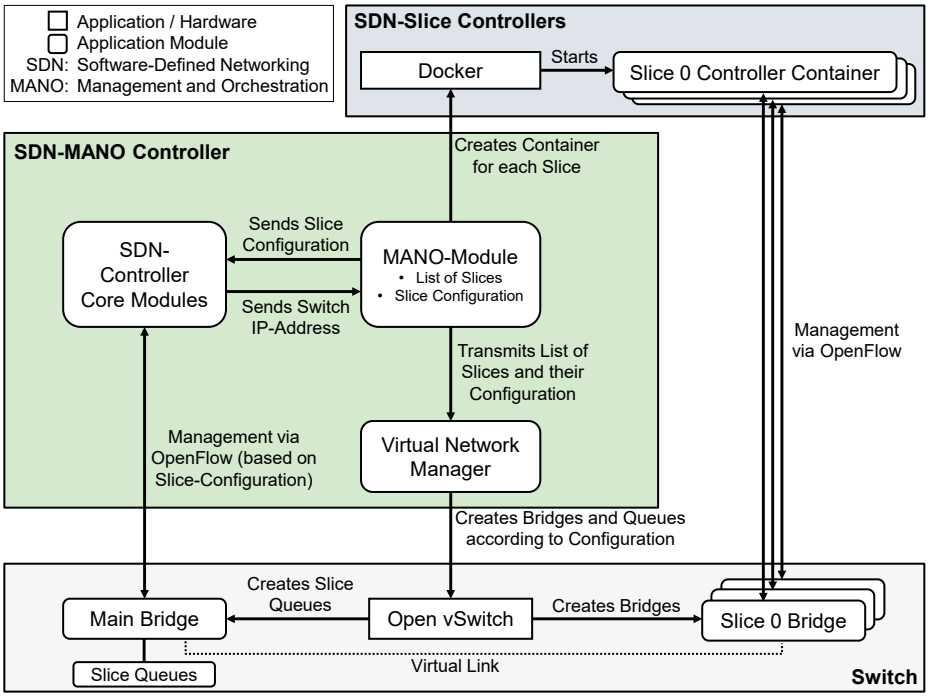


Fig. 5.33: The developed network slicing architecture.

The slice bridges comprise virtual ports, which are residing within the main bridge. The orchestration is done via the MANO controller, which dynamically instantiates slice controllers (e.g. via Docker), which in turn can be optimized for each application scenario. In the event of traffic entering the data plane, the MANO controller assigns packets to the respective bridge. There, the flow is mapped to the respective QoS queue and virtual destination port on the main bridge regarding the specific protocol or other criteria of the packet. This is done by the respective slice's controller. For each hop, the slice controller repeats this procedure of directing the flows to the main bridge. Unknown flows or not specified matches are handled on a best-effort basis. While this first part focuses on wired 5G communications, compatibility with the air interface slicing technologies presented in later sections.

5.4.2.2 Empirical Evaluation of 5G Core Network Slicing

Overview of the Testing Environment The testbed scenario is depicted in Figure 5.34. Six servers are assigned in pairs for each of the different use cases. These servers function as hosts to either send or receive data traffic over the sliced network. Furthermore, four machines are designated as SDN controllers, where three of them act as slice controllers running Floodlight and one is the MANO controller employing Ryu. For

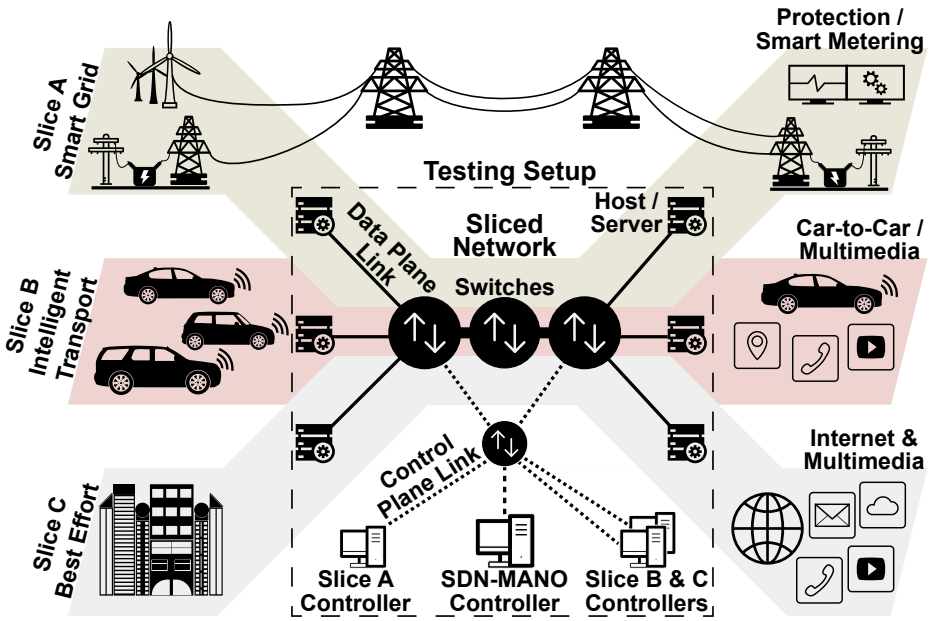


Fig. 5.34: Evaluation scenario in the testing setup.

measurement interference avoidance, three different networks, namely out-of-band control, maintenance, and sliced data-plane network, are in use. The Precision Time Protocol (PTP) is utilized to synchronize the controller clocks with a maximum deviation of 153 µs and a mean deviation of 16 µs. The underlying network load is generated via iPerf2 community edition and consists of User Datagram Protocol (UDP) packets. The maximum performance of the evaluated methods needs to be determined by considering the different layers of the ISO-OSI stack. The Ethernet frame size on the 2nd layer of the OSI model is 1512 B, which is used as a point of reference. Since performance evaluations are located on layer 4, the payload (i.e. goodput) results in 1470 B, which is 97.2 % of the layer 2 data rate. The following measurements were repeated at least 100 times with a minimum duration of 1 min per run.

Evaluation Scenarios Scenario A depicts a performance study, where key aspects from 5G and critical infrastructure communication are evaluated such as delay and data rate for varying network loads. Therefore, the overhead of our approach is determined to demonstrate the efficient use of resources. By using 100 Mbit/s Ethernet links, possible limitations can be avoided while simultaneously affording the option of CPU load monitoring during testing. Moreover, the independence of the slices from each other is verified, so any detrimental effects of errors or overload in one slice harming other slices can be precluded. Within the evaluation, the network load is increased in steps,

Tab. 5.1: Slices and traffic flows of the critical infrastructure communications scenario.

Slice (descending priority)	Use case	5G service class	Priority within slice	Hard min. data rate [Mbit/s]	Max. delay [ms]
Smart grids	Protection (IEC 61850)	uRLLC	Highest	50	1
	Smart metering	mMTC	High	200	20
Intelligent Transportation Systems	Floating car data	uRLLC	Highest	100	1
	Passenger Internet	eMBB	Low	450	10
Best-effort	Multimedia	None	Lowest	None	100

reaching beyond the maximum usable data rate/goodput, i.e. 97.2 % of the nominal layer 2 link capacity.

This approach represents cases in which end users try to use more resources than allocated for their respective slice, thus serving to demonstrate slice independence. The misconfiguration by operators of sliced communication networks is simulated as well. For this, two slices whose combined data rate exceeds the underlying physical network's data rate are configured. Scenario B represents a scalability analysis, where a viable approach for deployment in largescale, multitenant communication infrastructure is demonstrated. Since the number of slices should not influence the overall network performance, the delay performance for no, 2, 8, and 16 slices is analyzed. The available data rate is shared equally among the slices, with traffic streams utilizing 100 % of the respective slice's capacity. This ensures the exclusivity of side effects caused by slicing and not by network congestion or other factors. Furthermore, the validity of slice isolation and the stability of end-to-end delay is examined. For this, seven out of 21 slices are subjected to UDP-based traffic with data rates above the allocated limit. In contrast to scenario A, 1 Gbit/s Ethernet is used to stress test the concept. Finally, scenario C depicts a critical infrastructure communication including FCD of ITS and the IEC 61850 SG protocol.

Since both use cases are considered as uRLLC 5G services and therefore assigned the highest priority, the slices on which they are transmitted is allocated equal priority, including the dedicated SDN controllers. Smart metering (representing mMTC) and passenger internet (eMBB) are included in the related slices to demonstrate the ability of traffic distinction in our solution. Moreover, a best-effort slice is included for handling multimedia traffic and perpetually transmitting low-priority data at 950 Mbit/s, roughly consuming the maximum available layer 4 goodput of the 1 Gbit/s network. Therefore, if another slice needs a specific data rate, the network is overloaded and reallocated due to its differing priorities. The maximum tolerable delays for each priority data is given in Table 5.1. For the evaluation, the given data rates were achieved by bundling

multiple traffic flows. However, the bundling of constant data rates can be found in real-world use cases such as smart metering.

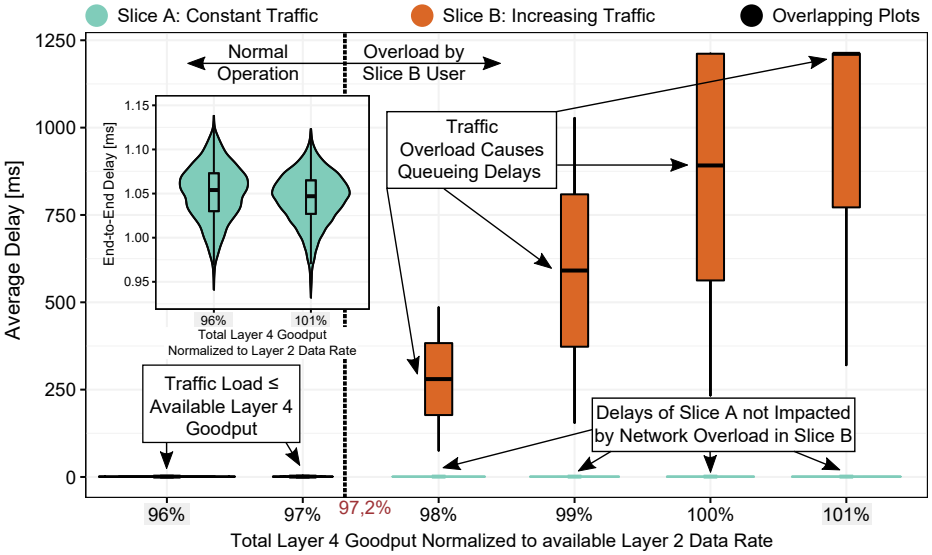


Fig. 5.35: End-to-end delays of two slices for varying traffic loads.

Evaluation Results In Figure 5.35 the end-to-end delays of two slices for varying traffic loads are depicted [360]. The physical 100 Mbit/s Ethernet network is shared fairly; slices A and B transmit UDP packets. Below the aforementioned limit of 97.2 %, the median end-to-end delay is located at 1.05 ms with a variance of approximately 0.05 ms. Nevertheless, when step-wise exceeding the limit at slice B, an overload situation is created resulting in increased delays. At 101 % load the median delays rise sharply up to 1.212 ms. However, the delays at slice A remain unaffected, even compared to no slicing as depicted with the enlarged violin plots. Hence, the isolation of the slices is shown. The misconfiguration by the operator is simulated and depicted in Figure 5.36.

Slice A receives a data rate of 40 Mbit/s (38.9 Mbit/s effectively on layer 4). The total sum of queue data rates (depicted on the x-axis) should not exceed the theoretical layer 2 limit of the 100 Mbit/s Ethernet link. This maximum is calculated as the ratio between frame sizes at layers 2 and 1, which amounts to 1512 B/1532 B = 98.7 %. In the event of misconfiguration, slice B tries to utilize resources, which do not exist. Therefore, slice B cannot maintain the layer 4 goodput. While slice A consumes the HTB tokens and remains stable, the data rate of slice B levels out to 56.9 Mbit/s, which is below the configured 60 Mbit/s queue data rate on layer 2. Overhead in terms of achieved throughput is not observed and therefore confirms expectations. Figure 5.37 depicts scenario B evaluation results.

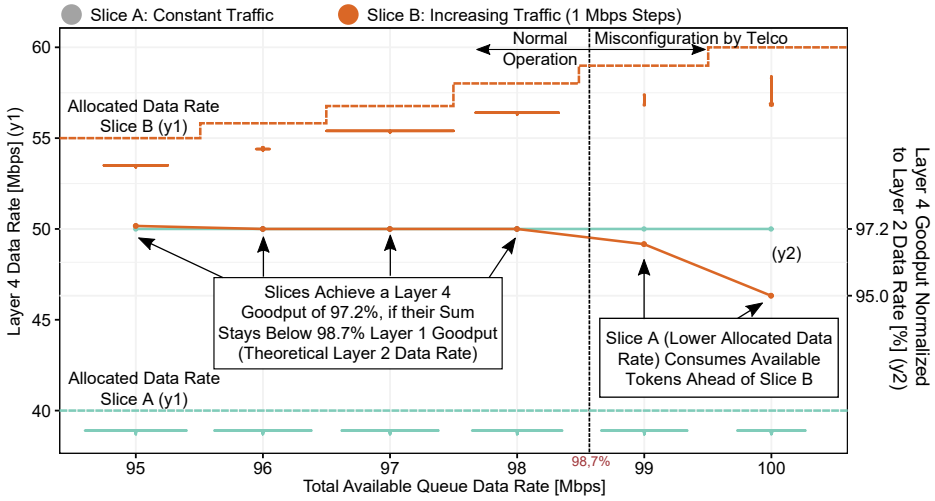


Fig. 5.36: Impact analysis of misconfiguration by the physical network operator.

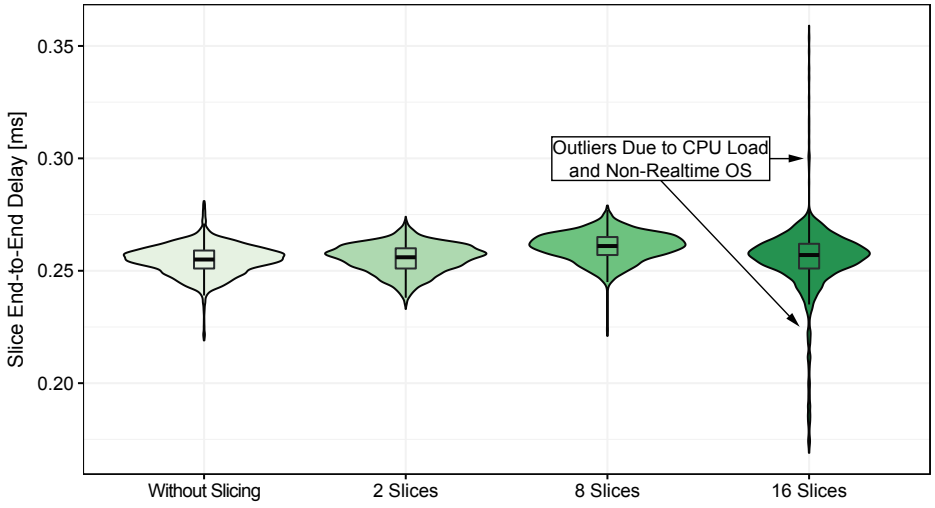


Fig. 5.37: Evaluation results of multiple slices on end-to-end delays.

The end-to-end delays of dedicated, sliced networks are given with an overall capacity of 1 Gbit/s. The data rate is fairly distributed between the slices with all of them in idle mode except one. With as many as sixteen slices, the delays remain stable. However, outliers of up to 0.36 ms may be a result of CPU context switches, which are required since the hardware provides a maximum of eight threads and queues. The outliers down to 0.17 ms are presumably caused by the non-realtime reduced timer/interrupt coalescing of the Network Interface Card (NIC) and Operating System (OS), which is triggered by the raise in computational load. Therefore, for highly sliced networks, performance optimizations of the developed source, real-time kernels and higher thread-count CPUs are to be pursued. The following stress test is given in Figure 5.38.

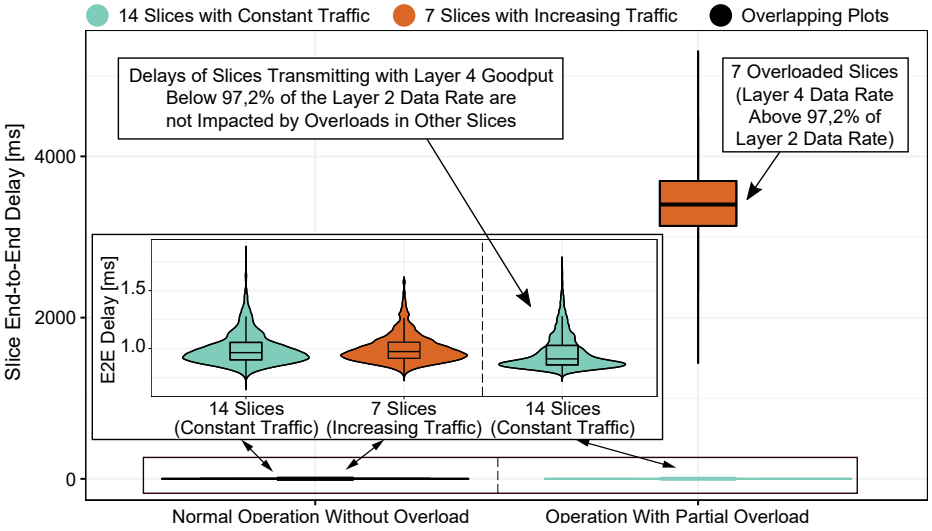


Fig. 5.38: Stress testing of scalability with partial overload.

Under normal operation (left-hand side), 21 coexisting slices can fully utilize their allocated data rates with a stable median delay of 1 ms. In comparison with previous tests, the delays are higher, because of more slices sharing a slower physical network of 100 Mbit/s. On the right-hand side, partial overload is simulated, resulting in seven slices trying to exceed their limits and accordingly causing increasing delays up to 3.5 s. However, the other slices stay unaffected. Therefore, the isolation of network slices remains equally robust even with high loads in several slices. Furthermore, the data rate remains stable across all realized scalability tests.

Finally, Figure 5.39 summarizes the measured data rate of the traffic flows given in Table 5.1. It starts with only one traffic flow of 950 Mbit/s, which fully utilizes the physical network on a best-effort basis. Therefore, even though the traffic continues throughout the test, network resources can be allocated to higher priority slices. Thus,

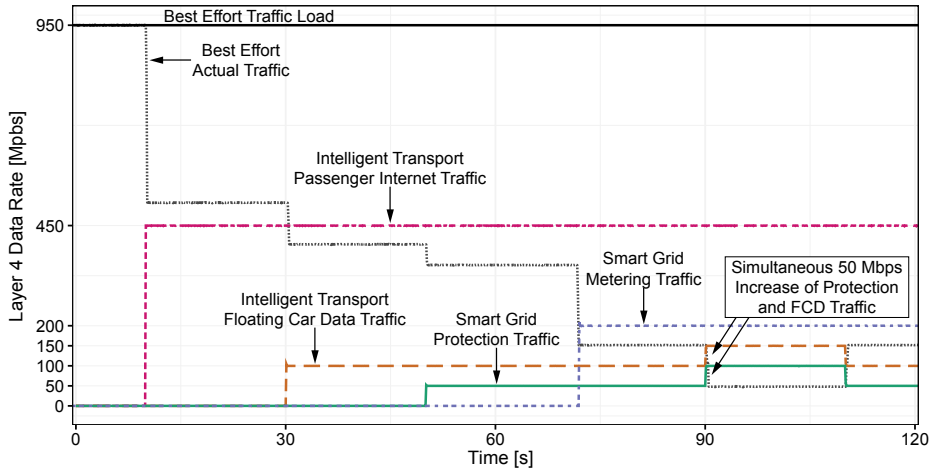


Fig. 5.39: Critical infrastructure communication scenario—data rate allocation and slice isolation.

when passenger Internet and FCD traffic of the ITS slice is generated, the best-effort throughput is reduced nearly instantaneously. The same happens when introducing protection and smart metering traffic on the SG slice. For an especially critical test case, the protection and FCD traffic is simultaneously increased to 50 Mbit/s at 90 s into the measurement. Figure 5.40 depicts the end-to-end delay of the slices.

As shown, hard service guarantees are provided during these transitions. Best-effort typically stays below the set boundary of 100 ms. Nevertheless, outliers of about 350 ms occur, which are induced by slice overloads. The delay for smart metering and passenger internet stays below 3ms with a median of approximately 1.3 ms and therefore satisfy service-level guarantees. The outliers result from the starting phase. During slice reconfiguration, the violins of protection and FCD traffic show slight delay variance, which does not affect the requirements since it stays mostly below 0.5 ms.

5.4.3 Data-Driven 5G Network Slicing in the Radio Access Network

In contrast to the previous sections, in which network slicing in the core network was discussed, the focus here is on the Radio Access Network (RAN). In this context, the data-driven aspect of network slicing becomes more important, as low-latency slices can be realized only via the prediction of emerging network traffic. This relation is described in the following subsections.

5.4.3.1 Introduction to Data-Driven Network Slicing

Previously, the three main service types eMBB, uRLLC, and mMTC were introduced. The balance between uRLLC slices and eMBB slices is particularly challenging to maintain

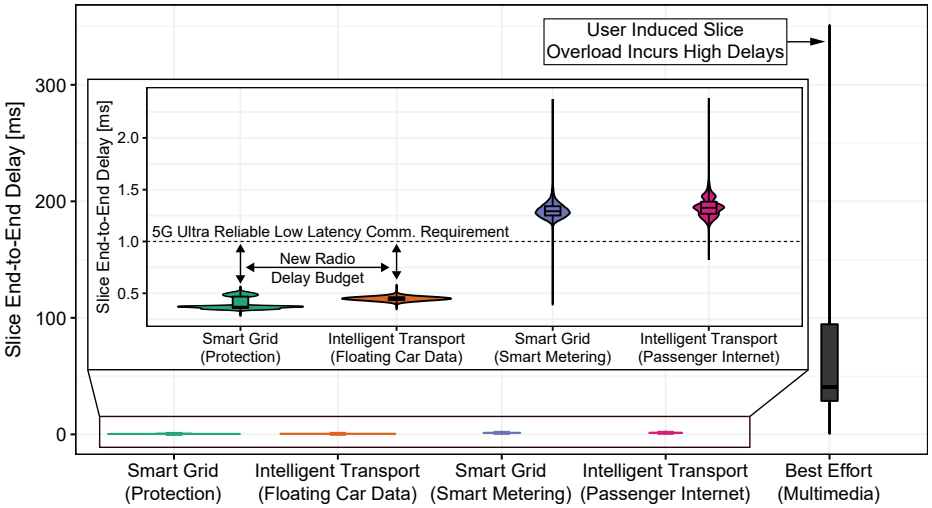


Fig. 5.40: Critical infrastructure communication scenario–end-to-end slice delays.

within Radio Resource Management (RRM), i.e. the network scheduler that is a crucial part of realizing network slicing within the RAN. To understand this relation, end-to-end latency components, which were derived from [493], are depicted in Figure 5.41.

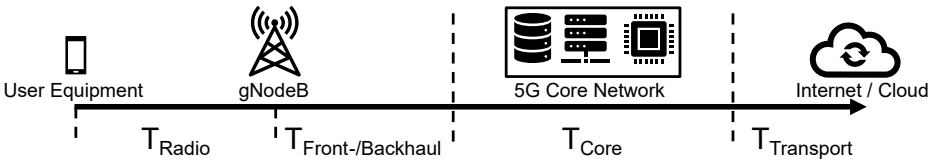


Fig. 5.41: Components in cellular networks which induce latency [493]. ©[2021] IEEE. Reprinted, with permission, from [54].

- $T_{Transport}$: Latency caused by the transmission of data through the transport network as when a web page is retrieved from the Internet.
- T_{Core}: Once data has been transmitted via the radio interface or from the transport network, it goes through the core network. This introduces additional latency, firstly because it is transmitted over an additional network, but also because the IP packets are unpacked and packed into different protocols required by the mobile network.
- $T_{Front-/Backhaul}$: The connection between gNodeB (5G base station) and core network introduces additional latency.
- T_{Radio}: The physical properties of the transmission channel are the main cause of radio latency, but the scheduler (T_{Sched}) also adds a significant delay.

The latency induced by the network scheduler (from here on T_{Sched}) is part of the T_{Radio} component, which is heavily reliant on the type of services or slices present in the 5G network. To further illustrate this, suppose there are two slices configured within the communication network, one uRLLC slice and one eMBB slice. From now on, the uplink direction of data transmission is focused (from UE to gNodeB). The UEs or the applications within the eMBB slice are data-rate intensive, which means that *as many Resource Blocks (RBs) as possible* are to be scheduled. By contrast, applications in uRLCC slices are not data rate intensive, but are to be scheduled *as fast as possible* to minimize the overall end-to-end latency (cf. Figure 5.41). This means that in order to minimize the scheduling latency, it is crucial to issue the scheduling grants before a request is even generated. For this, the so-called Configured Grant (CG) or proactive scheduling will be introduced in 5G [375]. As the name suggests the scheduling grants can be configured in advance to ensure Quality of Service (QoS) requirements . The major challenge, however, is that this requires a prediction of future data demands and channel qualities to allocate the required amounts of RBs for each network slice. The exact prediction of RBs for the uRLLC slice is crucial in this process because end-to-end latency will increase significantly if the predicted RBs are too low, which will induce retransmissions. If the predicted RBs are too high, the unused RBs will be wasted and thus not available for other network slices within the cell. This in order affects the aforementioned balance between uRLCC and eMBB slices, as the required prediction will induce prediction errors and thus waste resources for the data rate-intensive eMBB slices.

The remainder of this section will describe a data-driven CG-based scheduling and simulation framework called *SAMUS* [54], or *Slice-Aware Machine learning-based Ultra-reliable Scheduling*.

5.4.3.2 Description and Methodology of SAMUS: Slice-Aware Machine Learning-based Ultra-Reliable Scheduling - A Data-Driven Network Slicing Framework

5G-RGS (5G Resource Grid Simulation) Framework Figure 5.42 provides an overview of all modules, inputs, and outputs of the SAMUS system. As can be seen, the SAMUS system is not only comprised of the actual SAMUS scheduler prototype but additionally includes the 5G Resource Grid Simulation (5G-RGS) framework, which was specifically developed to evaluate the SAMUS scheduler prototype. There, channel conditions and data amounts of each User Equipment (UE) (or the external data used to predict the amounts) are provided as input to both modules. The 5G-RGS framework is then able to calculate resulting data rates and packet latencies ($\propto T_{Sched}$) based on the aforementioned channel conditions, the Transmission Time Interval (TTI) , as well as the allocated RBs. The last is a product of the data-driven SAMUS scheduler prototype, which generates CGs in the form of resource grid allocations based on external data, a process which will be described later in this section.

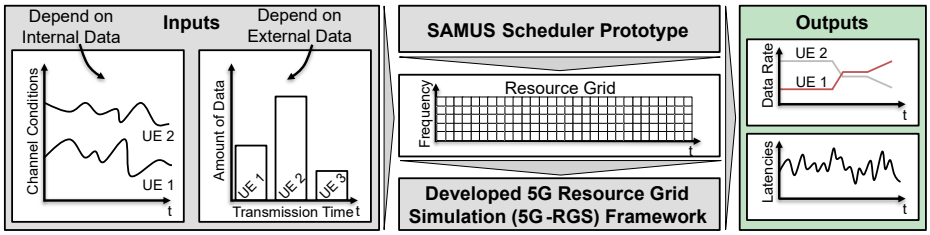


Fig. 5.42: SAMUS system overview including all modules, inputs, and outputs. ©[2021] IEEE. Reprinted, with permission, from [54].

In order to further detail the data rate calculation process of the 5G-RGS framework, Equation 5.15 is provided,

$$\textit{Data Rate (Mbit/s)} = 10^{-6} \cdot \sum_{n=1}^{N_{UE}} \left(\frac{TBS^{(n)}}{TTI} \cdot N_{TTI} \right) \tag{5.15}$$

where N_{UE} describes the UE amount of the slice, $TBS^{(n)}$ the Transport Block Size (TBS) available for the n-th UE in bit, and N_{TTI} describes how many TTIs are available in a second (the default here is the New Radio (NR) specification of 1 ms).
Moreover, packet latencies are calculated via Equation 5.16:

$$\textit{Latency (ms)} = (I_S - I_C) \cdot TTI \tag{5.16}$$

where I_S represents the scheduling interval of a final packet bit transmission and I_C the interval of packet creation. Note that latency components like retransmissions or other components of (T_{Radio}) are neglected. Based on scenarios in [53], the 5G-RGS framework was successfully validated.

SAMUS Scheduler Prototype As can be seen on Figure 5.43, the inputs for the SAMUS scheduler prototype are comprised of channel conditions or Channel Quality Indicators (CQIs) and data amounts of each UE or Buffer Status Reports (BSR), which are generated from historical data (hence, data-driven). Apart from the fact that data-driven (low-latency) CGs can be generated, traditional (latency-intensive) Scheduling Requests (SRs) can also be processed by the SAMUS scheduler. The CGs, if predicted correctly based on historical data, can reduce the scheduling latency T_{Sched} down to zero. For generating the RBs and the data-driven CGs, the ARIMA (Auto-Regressive Integrated Moving Average) method is utilized to predict the future traffic data demands and CQIs.

To ensure safe operation of mission-critical slices, resources of critical applications are allocated first, while granting the remaining RBs to best-effort (eMBB, no QoS) slices (based on the *Greedy Network Slicing Scheduler* in [53]). Traditional scheduling is used whenever a packet could not be transmitted due to prediction errors (effectively increasing scheduling latency). As a result, a resource grid in the form of a matrix is

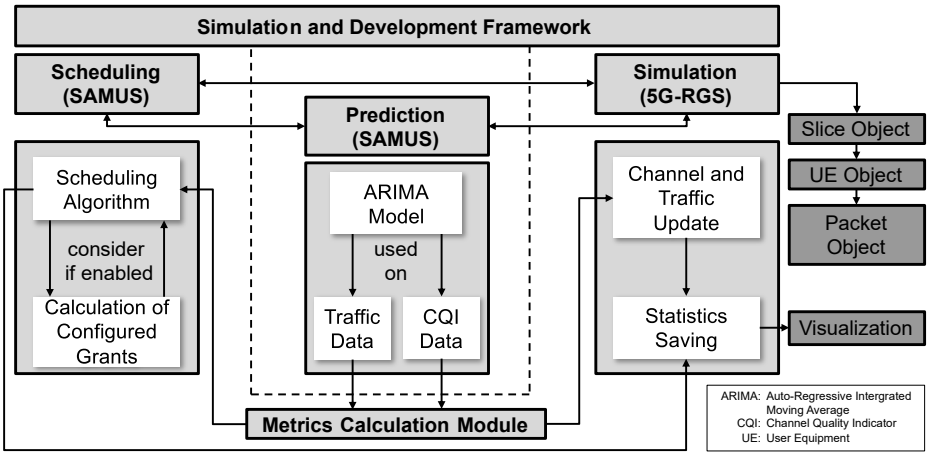


Fig. 5.43: Overall SAMUS framework including all modules and their interactions. ©[2021] IEEE. Reprinted, with permission, from [54].

passed on to the 5G-RGS, which calculates and protocols the Key Performance Indicators (KPIs), e.g., data rate and latencies. Finally, CQI and BSR values are updated and a new cycle is initiated.

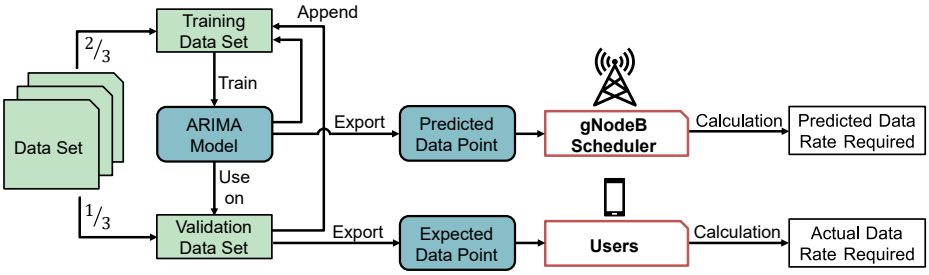


Fig. 5.44: Flow chart of SAMUS's prediction module utilizing ARIMA for training and operation. ©[2021] IEEE. Reprinted, with permission, from [54].

The details of the ARIMA-based prediction module are depicted in Figure 5.44 as a flow chart. There, the dataset associated with a slice is split up into training and validation datasets with a ratio of $\frac{2}{3}$ and $\frac{1}{3}$, respectively. Subsequently, the ARIMA model is trained in the course of offline learning based on the training dataset in order to predict future data, which in turn is utilized by the SAMUS scheduler to generate CGs. The data rate that corresponds with these CGs is the so-called data rate *predicted*. By contrast, the *actual* data rate required is calculated based on the validation dataset. The value for the actual data rate required results from the actually transmitted packets within the simulation, which are generated in order to test the prediction quality of the ARIMA

module. Also, online learning is facilitated to further optimize predictions based on the newly acquired data during the simulations.

These simulations, which were utilized to evaluate the SAMUS scheduler prototype are described in the next section.

5.4.3.3 Evaluation of the Data-Driven Scheduler Prototype SAMUS

Evaluation Scenario and Parameters As indicated in the previous sections, the SAMUS framework was evaluated based on a realistic network slicing scenario . In order to be able to compare the novel approach of the SAMUS scheduler to traditional methods as well as to present different trade-off strategies, so-called *modes* were designed and utilized.

The following modes were configured and evaluated:

- *Mode 1*: Traditional scheduling with request and grant method (without CGs)
- *Mode 2*: Fixed amount of RBs (fixed CGs)
  - *Mode 2.1*: **Average** historical data rate used as amount of fixed grants (fixed **optimistic** approach)
  - *Mode 2.2*: **Maximum** historical data rate used as amount of fixed grants (fixed **pessimistic** approach)
- *Mode 3*: **Data-driven** CGs (predicted based on ARIMA)
  - *Mode 3.1*: Predicted CGs as is (**No over-provisioning**)
  - *Mode 3.2*: *Over-provisioned* predicted CGs (**With 10 % over-provisioning**)

The mode configuration as well as other simulation parameters like configured and simulated network slices are listed in Figure 5.45.

The three realized slices in the evaluation scenario are also depicted in Figure 5.46, which are defined as follows:

- *Smart Grid (SG) slice (uRLCC - Highest priority)*: The *Smart Grid* slice is modeled after photovoltaic systems transmitting data to regulate energy generation. The National Renewable Energy Laboratory (NREL)[2] provides open data for solar activity, which is used to train the ARIMA model and generate data traffic proportional to the solar activity.
- *Electric Vehicle (EV) charging slice (uRLLC - High priority)*: EV charging-point occupancy data of the German city Bonn[3] was gathered and data traffic based on this dataset is generated for the EV charging slice.
- *Best-Effort (BE) slice (eMBB - Low priority)*: A constant rate of 18.96 Mbps is generated, which corresponds to the remaining capacity of the cell, to simulate devices with high data rate demands and to measure the remaining data rate within the

2 See https://www.nrel.gov/grid/solar-power-data.html.

3 See https://new-poi.chargecloud.de/bonn.

General Settings

Channel Bandwidth	20 MHz
5G Subcarrier Spacing	15 kHz
Channel Quality	Fixed Modulation and Coding Scheme (MCS) of 15
5G MCS Index Table	64QAM
SR Occasion	every 4 ms
Packet TTI	1 ms
Simulated Time	1 h

Slice-Specific Settings

Smart Grid (uRLLC)	Electric Vehicle Charging (uRLLC)	Best Effort (eMBB)
5 UEs	4 UEs	2 UEs
variable aggregated throughput	variable aggregated throughput	18.96 Mbps aggregated throughput

Mode-Specific Settings

Mode 1	Mode 2.1	Mode 2.2	Mode 3.1	Mode 3.2
No Configured Grants (CG)	Fixed CGs (optimistic)	Fixed CGs (pessimistic)	Predicted CGs	Predicted CGs
No Overprovisioning (OP)	No OP	No OP	No OP	10% OP

Fig. 5.45: Settings and parameters of the simulation framework and the different modes utilized in the evaluation of the SAMUS framework. ©[2021] IEEE. Reprinted, with permission, from [54].

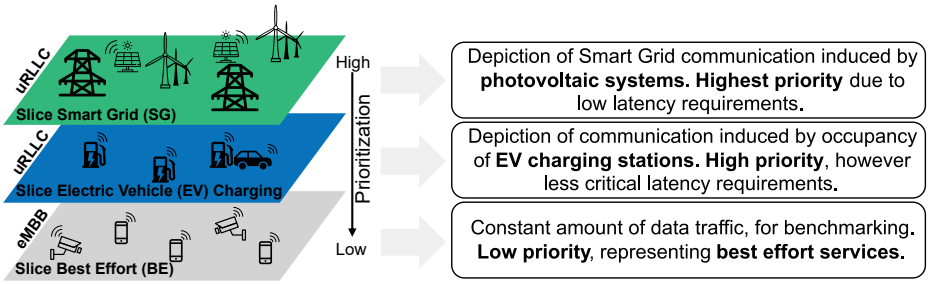


Fig. 5.46: Evaluation scenario comprising mission-critical and best-effort network slices to analyze different trade-off strategies within the SAMUS system. ©[2021] IEEE. Reprinted, with permission, from [54].

non-critical eMBB slices, after mission-critical slices are served by the SAMUS scheduler.

In the following section, the evaluation of the SAMUS framework is presented based on this scenario.

Evaluation Results The 5G-RGS framework described before was utilized to evaluate the SAMUS scheduler prototype based on the modes presented earlier, which represent different trade-off strategies between the balance of uRLLC latency and the eMBB data rate. For this, a 60 min interval was analyzed (cf. Figure 5.45), which represents a time frame of highly dynamic activity within the different slices such as the time of sunrise in the SG slice or the time of rush hour in the EV slice.

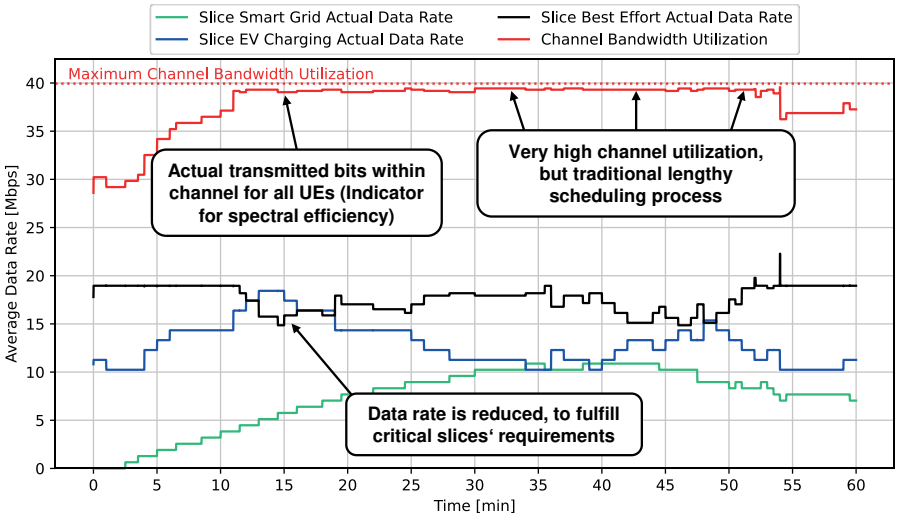


Fig. 5.47: Data-rate progressions for the different network slices in mode 1 (5G parameters only–traditional scheduling). ©[2021] IEEE. Reprinted, with permission, from [54].

In Figure 5.47, the results for mode 1 are depicted, where the average slice data rate in Mbit/s is plotted as a function of the simulation time in min. The dotted and solid red lines represent the maximum and the actual channel bandwidth utilization, respectively. This indicates the efficiency of resource usage, i.e., high channel utilization means low RB wastage. The green, black, and red solid lines represent the average uplink data rate transmitted for the SG, EV charging, and Best-Effort (BE) slices, respectively. Based on the *Greedy Network Slicing* method, it becomes clear that the available RBs are allocated to the higher priority slices at the expense of the BE data rate. This is the

desired behavior of the utilized traditional scheduling requests and grants, because RBs are distributed exactly as required and no resources are wasted. However, the main disadvantage of this approach is that it leads to very high scheduling latency (T_{Sched}). This connection becomes more clear when looking at Figure 5.48.

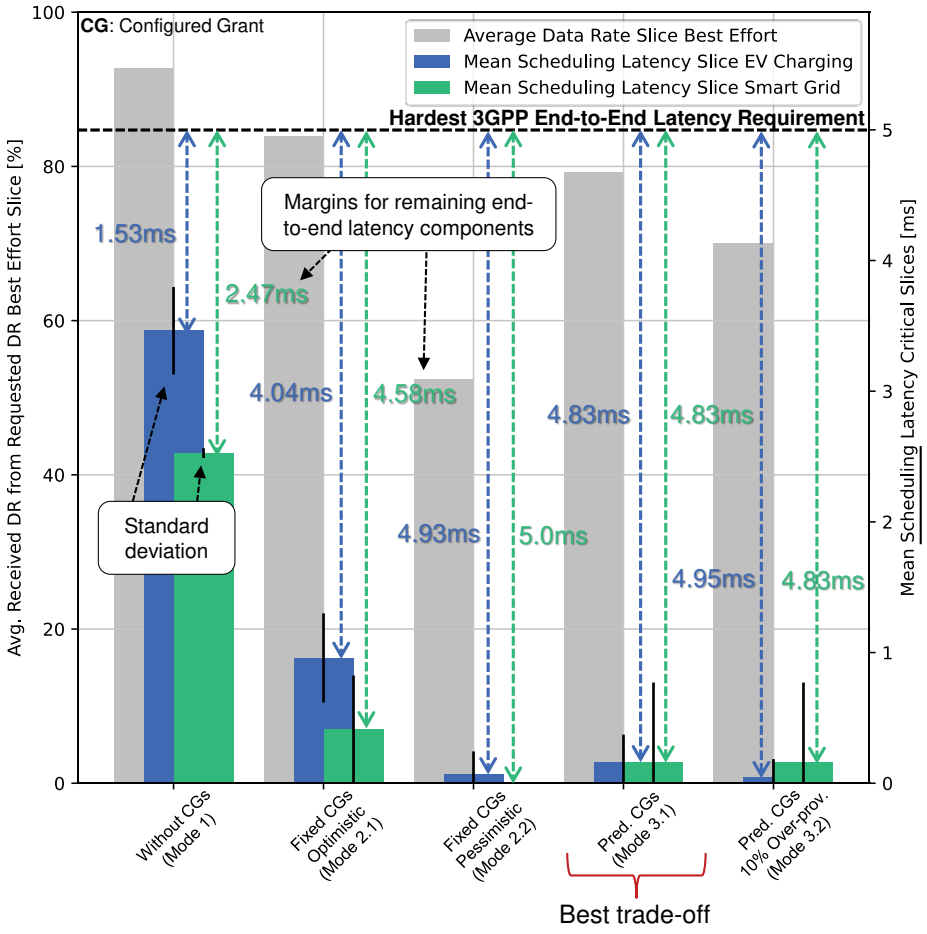


Fig. 5.48: Average BE data rates versus mean and standard deviation of high priority slice latencies (averaging window of 2 s and hardest 3GPP latency requirement according to 3GPP 23.501 [3]) for all modes. Margins for remaining latency components are indicated by the arrows in the respective colors of the slices. ©[2021] IEEE. Reprinted, with permission, from [54].

In this figure, two different y-axes are depicted describing the ratio of the average transmitted data rate from the actual data rate required by the BE slice on the left axis (gray bar plot) as well as the mean scheduling latency (T_{Sched}) of the high priority slices

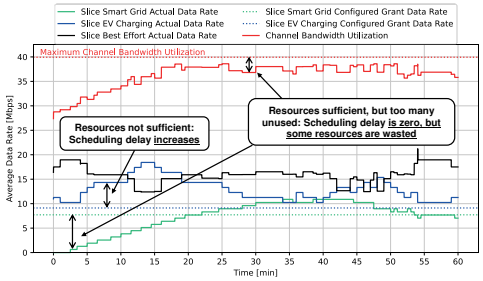
on the right axis (green bar: SG slice; blue bar: EV charging slice; black lines: standard deviation), respectively. The different modes are listed on the *x*-axis. By looking at the results for mode 1, it can be seen that the high bandwidth utilization represented by the ratio of transmitted data to required data is very high at 92.74 %. At the same time however, only 1.53 ms and 2.47 ms margins for other latency components based on the hardest 3GPP end-to-end latency requirements [3] are left for the EV charging and SG slices, respectively. This results from the utilization of lengthy traditional scheduling mechanisms.

For comparison, the results of the modes in Figure 5.49 can be consulted. As for the modes 2.1 and 2.2, depicted in Figure 5.49a and Figure 5.49b, the channel bandwidth utilization drops for both approaches, especially for the pessimistic approach. This is the result of the fixed allocation of RBs to the mission-critical slices. However, the effect of this method on the latency becomes clear again with a look at Figure 5.48. There it can be seen that the margins for the end-to-end latency, especially for the pessimistic approach, increase to almost 5 ms, since the scheduling latency drops to almost zero due to the constant availability of resources. By contrast, the data rate efficiency of the BE slice drops down to 52.22 %. Thus, the fixed CGs represent a very latency-focused approach, whereas mode 1 maximizes channel utilization.

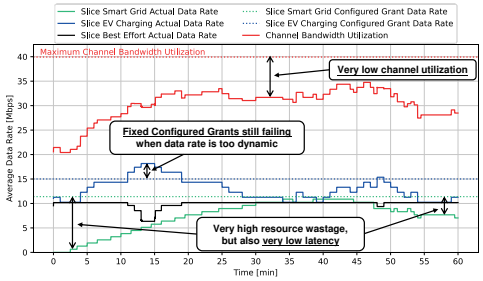
The data-driven ARIMA-based mode 3, which is the major contribution of the SAMUS framework, represents a good balance between these two extremes, as can be seen by looking at the data rates in Figure 5.49c, 5.49d and the latencies in Figure 5.48 for modes 3.1 and 3.2, respectively. Moreover, the channel bandwidth utilization is relatively high with an almost 80 % ratio of actual to requested data rate within the BE slice. Additionally, as data amounts of the mission-critical slices can be predicted very well, and thus, data can be instantly transmitted, margins for other latency components of 4.83 ms to 4.95 ms can be observed. The scheduling latency is zero most of the time.

5.4.4 Conclusion

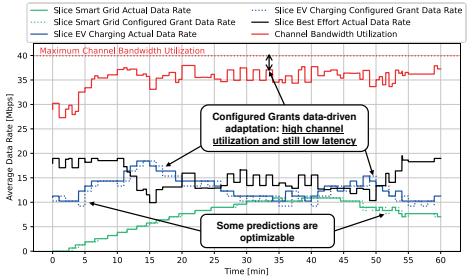
In this section, we presented *5G network slicing* approaches for both the core network and the radio access network. While the same goal is pursued in both domains, the implementation is all the more differentiated. Especially in the RAN, machine learning-supported methods will be indispensable, since the prediction of upcoming data traffic is a prerequisite for implementing low-latency slices, while still maintaining high spectral efficiency. This relation was shown here in this section on the basis of our SAMUS approach, which is able to efficiently trade-off resources between different slice types.



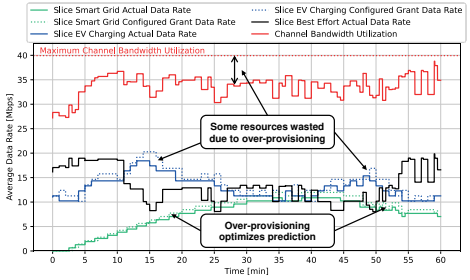
(a) Mode 2.1: Optimistic Fixed CGs



(b) Mode 2.2: Pessimistic Fixed CGs



(c) Mode 3.1: Over-Provisioned Predicted CGs



(d) Mode 3.2: 10 % Over-Provisioned Predicted CGs

Fig. 5.49: Comparison of Data Rates Within the Defined Slices and For All Modes Based on the Defined Evaluation Scenario. ©[2021] IEEE. Reprinted, with permission, from [54].

5.4.5 Acknowledgments

In addition to the CRC 876, part of his work has been supported by the Ministry of Economic Affairs, Innovation, Digitalisation and Energy of the State of North Rhine-Westphalia (MWIDE NRW) along with the Competence Center 5G.NRW under grant number 005-01903-0047 and by the Federal Ministry for Economic Affairs and Energy (BMWi) in the course of the project 5Gain under the funding reference 03EI6018C.

5.5 Potential of Millimeter Wave Communications

Karsten Heimann
Simon Häger
Christian Wietfeld

Abstract: For mobile communication networks, radio spectrum resources have always been a scarce commodity. With the cultivation of *millimeter Wave (mmWave)* wavelengths, a vast amount of spectrum at frequencies above 24.25 GHz has become available to serve the demands of *enhanced mobile broadband* services and applications of fifth-generation of mobile communications (5G). However, the higher carrier frequencies compared with the heretofore allotted spectrum comes with novel challenges for the operation of a cellular network: The more significant propagation losses require directional/beam antennas and their directivity needs to be adjusted permanently and individually per user. In addition, the poor obstacle penetration necessitates a careful beam alignment based on Line-Of-Sight (LOS) conditions. In case of obstructions, signal reflection paths need to be leveraged, which may be volatile and time-consuming to discover. By means of signal quality measurements, a self-contained beam tracking may maintain the LOS or virtual LOS via reflections to mobile devices. As a further feature, the directional knowledge of the base station antenna beams can even be exploited for a bearing-like localization approach allowing for an enhanced network positioning service compared with cell-level approaches. The sophisticated Software-Defined Radio (SDR)-based mmWave platform allows for the experimental evaluation of the mentioned features. The results prove the potential of mmWave communications for various vehicular and logistics use cases. The lessons learned will go into future research directions such as smart radio environments. The novel technology of Reconfigurable Intelligent Surface (RIS) is a promising strategy for improving the capabilities of the general environment to supply better radio conditions to a wireless channel in non-LOS conditions. For example, a RIS can purposefully redirect the base station's mmWave pencil beam to reach a device in an obstructed area and thus extend the network coverage. Future integrated, radar-like sensing capabilities of communication networks are expected to operate at mmWave frequencies due to large bandwidth, high directionality, and low multipath features promising high-quality measurements. We show that the channel information of current mmWave systems, beam orientation in particular, already enables novel sensing applications.

5.5.1 Introduction

Besides the growing number of Internet-of-Things (IoT) devices with low data rate but high coverage requirements and applications demanding reliable or low-latency data transfer, another main direction of impact of fifth-generation mobile networks (5G) focuses on the enhanced Mobile Broadband (eMBB) services. It is believed that future applications such as augmented, virtual, or extended reality necessitate a high-performance wireless network infrastructure.

While optimizing the utilization of the traditional *sub 6 GHz* radio spectrum in terms of spectral efficiency, this resource is already heavily used. However, with 5G the third Generation Partnership Project (3GPP) targets additional spectral resources in the mmWave domain (particularly from 24.25 GHz to 52.6 GHz in Frequency Range 2 (FR2)) [2, Table 5.1-1]. Frequencies in the THz domain will also be targeted in future mobile networks promising even larger bandwidths–and use thus a vast amount of resources. Although these resources ought to enable an enhanced throughput at the air interface, novel challenges arise due to the higher frequencies.

Unlike the popular misconception, the more severe path loss itself is not the main issue, because higher frequencies allow for an increased antenna gain within the same space constraints. With this, the path loss itself is more than compensated. Nevertheless, the increased antenna gain is achieved by a more distinct directivity, which demands a proper antenna alignment. Phased Array Antennas (PAAs) resolve that issue by interconnecting multiple antenna elements, so that a sophisticated superposition of the processed signals allows for an adjustable radiation characteristic known as *beamforming* or *spatial filtering*. A PAA applies phase shifts to the signals of the individual antenna elements. For example, at a Uniform Linear Array (ULA), N antenna elements are uniformly spaced with some distance d (mostly at half a wavelength, so $d = \frac{\lambda}{2}$). To create a beam directivity that points towards a direction θ, phase shifts of 0, ϕ, 2ϕ to $(N-1)\phi$ are applied to the respective antenna elements 0, 1, 2 to $N-1$ with $\phi = \frac{2\pi}{\lambda} d \cos\theta$ [39, Chapter 6]. Put simply, the number of elements N determines the beamwidth and antenna gain. In general, a larger N leads to a higher gain and a more focused beam.

This means that a steerable directivity is feasible and can be achieved electrically or by software. The transmitter antenna's *beam* can be dynamically aligned to a receiving antenna and vice versa facilitating radio propagation by high transmit-and-receive antenna gains. However, radio signaling as part of the control plane of the Radio Access Network (RAN) needs to carry out this alignment task in a timely manner, which could be challenging due to the volatile radio conditions and the users' mobility.

For example, the alignment could be performed by means of a potentially time-consuming discovery procedure such as *beam sweeping*. The coverable angular space is iteratively sampled by switching the beam through different pointing directions. In doing so, the beamwidth constitutes a trade-off between a higher gain and a reduced number of iterations required to sample the complete angular space. While a precise

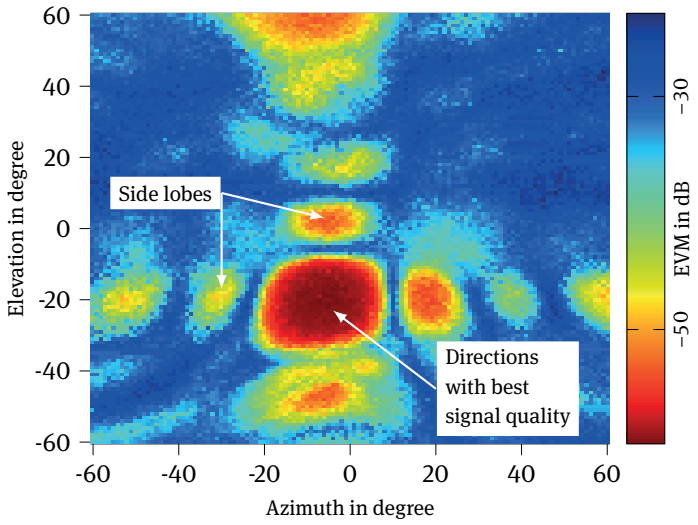


Fig. 5.50: Exemplary heatmap illustration of signal quality measurement during an exhaustive search *(beam sweep)*. The signal quality is given as Error Vector Magnitude (EVM) with lower values representing better signal qualities. The red area represents the beam-pointing directions with a suited signal quality. ©[2020] IEEE. Reprinted, with permission, from [268].

beam alignment is generally feasible in the analog domain, a number of quantized main lobe/beam-pointing directions as large as the number of antenna elements is often used to span an angular grid, where the selectable beams have the least possible overlap [148, Chapter 6].

Although the exhaustive sweep procedure can be accelerated by using multiple beams in parallel, each beam requires its separate RF-chain which are expensive with regard to their costs and energy demands. For this reason, it is believed, that analog or hybrid beamforming, where only a small number of parallel beams is available, is applicable for mmWave communications.

During a sweep, the measurements of the signal quality can be interpreted as a *heatmap*. Figure 5.50 depicts such a heatmap, with the most-suitable directions represented by the red spots. In a mobile network like 5G, the base station continuously transmits some reference signals at different beam directions in the downlink, so the User Equipment (UE) is able to select the strongest one, while performing a sweep with its receiving beam. Since such systems are defined for Time Division Duplex (TDD), channel reciprocity can be assumed and the UE can use the determined beam configuration for initially accessing the network and reporting back the suited beam direction pair. These directions can subsequently be used for further transmissions/receptions until the mobile device has moved or some obstruction occurs, which means the measured beam—dependent signal quality becomes outdated.

5.5.2 Beam Tracking for Interruption-Free High-Performance Communications to Mobile Devices

A proper beam alignment is as crucial to establishing a communication link as it is to maintaining it by tracking the mobile UE. The main drawback of the exhaustive search is its large search space and exploration time. Also, during this exploration, the beam points in various directions with weak signal quality, which may lead to a heavily reduced radio link performance or even a connection loss. For this reason, other procedures take into account a position or previous direction information and potentially the device mobility to facilitate an interruption-free utilization of the radio resources for purposeful data transmissions. This means, that once a proper alignment is initially discovered, *beam tracking* is preferably applied to follow the device movement. Only in case of a connection loss due to, say, sudden blockage, another comprehensive exploration might be required for radio link recovery.

In our works [269, 270], we analyze the applicability of beam tracking for supplying mobile users with mmWave radio links.

As a proof of concept, the position of a mid–flight drone/Unmanned Aerial Vehicle (UAV) is recorded by an optical reference system allowing for a geometry-based, precise calculation of the required beam-pointing direction. Figure 5.51 gives an overview of the experimental setup. The UAV movement describes an arc at a fixed distance of 1.8 m from the stationary active antenna/PAA. The central experiment logic controls this movement, processes the UAV position, sends corresponding beam-pointing commands and logs the measured performance indicators such as signal quality and data rate. In addition to the PAA's beam alignment, the passive horn antenna at the UAV can be aligned horizontally by means of the UAV's yaw rotation.

The evaluation results are condensed in the time-series graphs of Figure 5.52. When only the yaw rotation of the UAV is used to align the passive antenna at the UAV, the communication link is active only within a small range around the center direction, which is where the PAA is configured to point at in the static case. On the contrary, when only the PAA's pencil beam is continuously aligned towards the UAV, the misalignment of the horn leads to connection losses. Since the horn has a wider beamwidth, the tolerance for a misalignment is larger. Finally, when both transmitter and receiver antenna are continuously aligned to each other, a stable link is observed in terms of a constantly high data rate of about 2.8 Gbit/s. This proves the general applicability of mmWave communications utilizing PAAs for beam alignment in scenarios with mobile users.

Since external position knowledge might not always be available and could require an additional, beam-alignment independent control link (for example at a conventional sub-6 GHz band) for reliable reporting, a self-contained beam tracking approach based on signal quality measurements is evaluated in [270]. Besides keeping the beam at a direction with a still acceptable signal quality, better beam-pointing directions need to be explored during a temporary impairment of the signal quality. In general, there is a

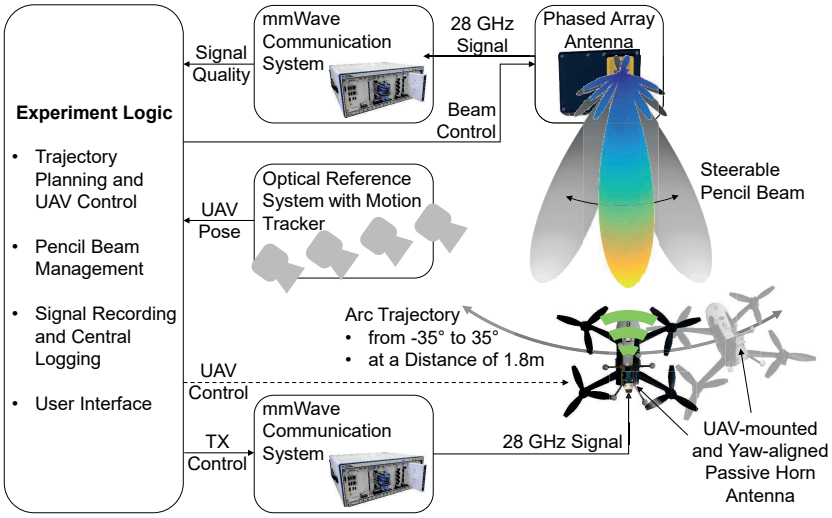


Fig. 5.51: Experimental setup for real-time mmWave beam alignment studies with a flying UAV based on [269]. The motion capture system provides position information of the UAV, which is processed to a beam-pointing direction and sent to the PAA as a control command.

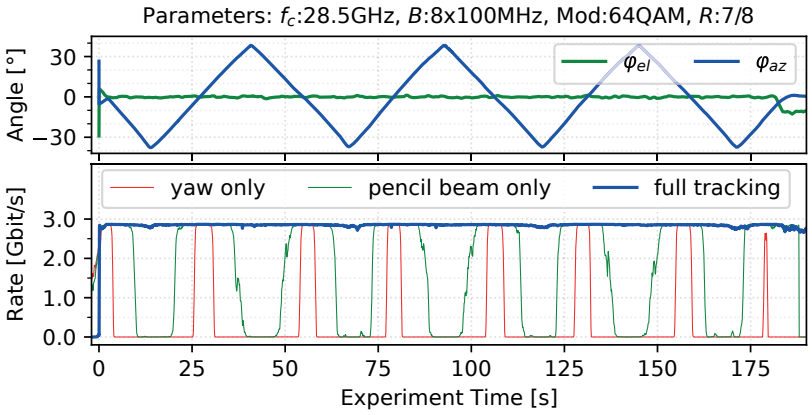


Fig. 5.52: Evaluation of mmWave beam alignment with a mid-flight UAV based on [269]. While connection losses occur without updating the directivity, a seamlessly high data rate can be achieved by tracking the movement of the mid-flight UAV with both transmitter and receiver antennas.

trade-off between the detail of exploration and the perceived signal quality by leveraging the aligned beam's gain, since the beam needs to be intentionally misaligned to explore the device's moving direction. Assuming that the device motion tracking under Line-Of-Sight (LOS) conditions requires only gradual changes in the beam-pointing direction, the search space can be substantially limited to the adjacent directions. For example in [270], a 3×3 grid in the azimuth and elevation plane of the angular space of beam directions is spanned centered at the last acceptable signal-quality direction. In doing so, every scanning cycle consists of as few as nine signal quality measurements and the subsequent search grid is centered on the direction with the highest signal quality. Although this appears to misalign beams in most cases, the small amount of measurements per cycle allow for low grid spacing below the beam width as long as the sample rate is significantly higher than the device's relative angular velocity. With this, the connection can still be maintained during the exploration. The minor reduction in antenna gain due to the slight misalignment can be compensated for by the communication system. For an experimental evaluation of this approach, the device motion is emulated in a reproducible fashion with a precise reference by using a rail system.

The statistical results of this empirical analysis is depicted as violin plots in Figure 5.53. While the signal quality is represented by the Error Vector Magnitude (EVM), where a lower value corresponds to a higher signal quality, the link performance is evaluated in terms of data rate. The emulated mobile device velocity is converted to the related maximal tracking dynamics from the antenna's perspective. A small exploration grid spacing of $\Delta = 1^\circ$ reduces the decline of the antenna gain due to a reduced misalignment only at low relative velocities, since this step size is not sufficient to keep track of the motion at higher dynamics. A larger spacing of $\Delta = 5^\circ$ deteriorates the link performance or may even lead to connection losses due to severe beam misalignments during exploration. Finally, the grid spacing needs to be fitted to both the device's velocity and the antenna's beamwidth. In the conducted test setup, a grid spacing of 3° empirically turned out to be a reasonable tradeoff. The results of laboratory evaluation thus prove, that a high–performance communication link can be maintained even for considerable device velocities.

With respect to the utilization of the novel radio resources at the mmWave domain, this beam tracking approach allows for efficient utilization of the spectrum by reducing the link outage due to lengthy exploration phases. In addition, the directional transmissions via *pencil beams* facilitate dense spatial reusability of these resources, since the interference within the mobile network is reduced.

As an outlook, future beam tracking techniques may incorporate reinforcement learning approaches to solve the exploration–exploitation tradeoff dilemma between a comprehensive exploration of beam-pointing directions with their associated signal qualities and a perfectly aligned beam with ideal signal quality conditions for data transmissions. In doing so, dynamic and reactive adjustments of the search space

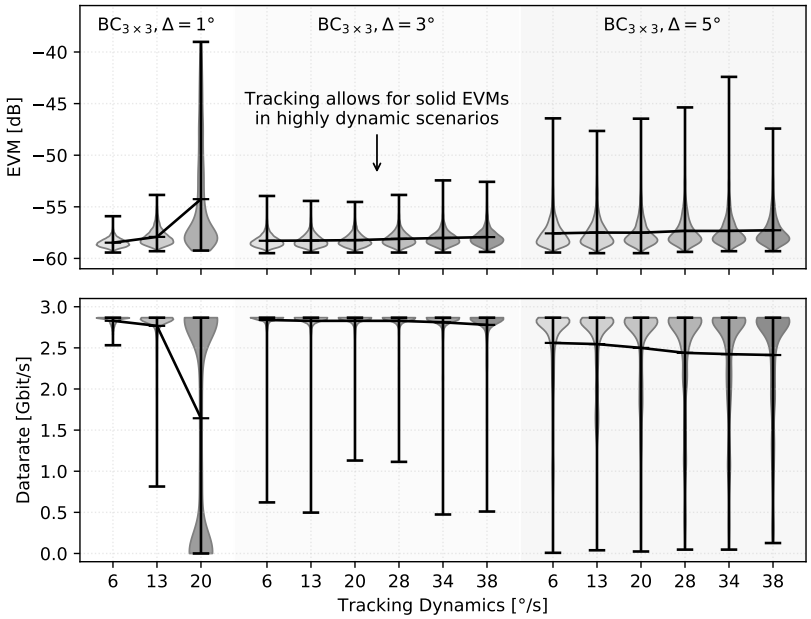


Fig. 5.53: Statistical beam-tracking evaluation. For exploring grid spacings Δ of 1°, 3° and 5°, the signal quality in terms of Error Vector Magnitude (EVM) and the link performance in terms of data rate is analyzed under different tracking dynamics. The tracking dynamics correspond to different mobile device velocities that are reproducibly emulated by a rail system. The constantly low EVM values at $\Delta = 3^\circ$ constitute a reasonable configuration with a stable link performance even at the highest tracking dynamics. ©[2019] IEEE. Reprinted, with permission, from [270].

(grid shape and spacing, for example) are conceivable according to the anticipated movement of the device.

5.5.3 Dual-use of Beam Alignment Information for Positioning of Mobile Devices

Although a proper beam alignment embodies a new challenge to mobile networks, once gathered the direction information could also be used for a bearing-based positioning service, as addressed in [268]. Conventionally, cellular network-based positioning utilizes signal-strength measurements in conjunction with propagation loss models, signal travel time, or propagation delay measurements (such as those used for the *timing advance* mechanism) for distance-based positioning or *lateration*. With the necessity for directional transmissions, angle-based methods utilizing direction information as a bearing are conceivable at the mmWave domain. By means of two or more intersecting bearings and known base station positions, a user-position estimate can be provided on

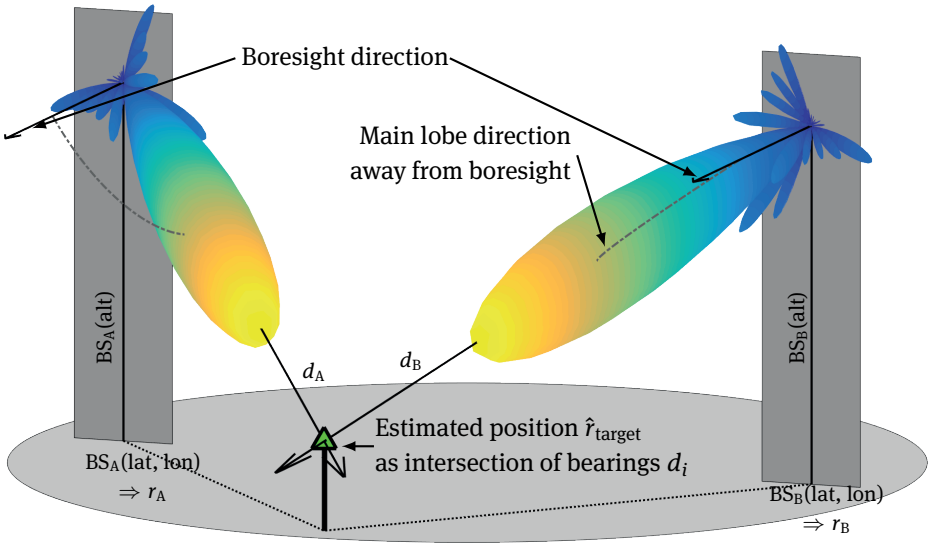


Fig. 5.54: A cross-bearing-based positioning utilizing mmWave beam alignment information. With known base station positions and beam directions as bearings, the mobile device position can be estimated as the intersection of the bearings. ©[2020] IEEE. Reprinted, with permission, from [268].

top of the ongoing wireless communication. In doing so, the accuracy strongly depends on the distance or constellation and the resolution of the direction finding.

Figure 5.54 illustrates the basic concept of this approach, which derives its origins from sea travel's cross-bearing. The estimated position $\hat{r}_{\text{target}}$ is defined as the position vector that minimizes the squared distance to the (two or more) lines spanned by the base station position vectors r_i and its beam-pointing direction d_i as direction vector. As a result, the position estimate is given as least squares approximation (with I as identity matrix):

$$\hat{r}_{\text{target}} = \left(\sum_i I - d_i d_i^T \right)^{-1} \left(\sum_i (I - d_i d_i^T) r_i \right) \quad .$$

From an exhaustive sweep, the direction vector d_i is estimated based on the beam-pointing direction with the highest signal quality. Since the area with reasonable signal quality turns out to be rather flat but noisy, this estimate is rather imprecise. For this reason, the centroid of the red region, which contains the highest signal quality, is taken as the improved direction estimate. In addition to the geometric location and orientation of the base station antennas, systematic deviations between the commanded and the actual pointing direction of the antenna beams are compensated. The experimental evaluation indicates the applicability of this approach and is depicted in Figure 5.55. In general, within the laboratory setup, a 3D Euclidean positioning error in the centimeter range is observed. The post-processing compensation for systematic deviations further

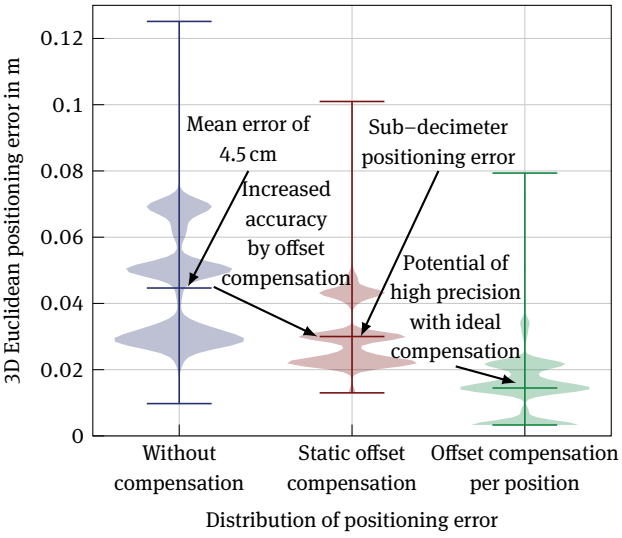


Fig. 5.55: Statistical evaluation of the positioning performance based on laboratory experiments. The 3D Euclidean positioning error lies in the centimeter range and can be further reduced by compensating systematic deviations in the direction estimates towards the UE. ©[2020] IEEE. Reprinted, with permission, from [268].

improves the estimated position and thus illustrates the potential of bearing-based mmWave positioning. Further details about the experiments can be found in [268].

The direction exploration as well as the post-processing compensation may be subject to machine learning-based optimization techniques introducing automated trade-off decisions between the resource utilization and positioning precision during runtime.

Nevertheless, hybrid procedures could combine distance measurements and bearings for a further enhanced positioning service of mobile networks. The large available bandwidth at the mmWave domain could be attractive for pseudoranging or Time Difference of Arrival (TDOA) considerations. The application of TDOA-based positioning utilizing the Ultra-Wideband (UWB) technology is analyzed in more detail in Section 3.5.

Due to the challenging propagation characteristics at the mmWave domain, a dense deployment of mmWave base stations is required and may lead to an enhanced system performance by utilizing approaches such as Coordinated MultiPoint (CoMP) or Dual Connectivity (DC), so the connection of one UE with multiple base stations at a time. Within a mobile network, the proposed positioning mechanism can be applied for both the downlink as well as the uplink direction. While the Angle of Departure (AoD) of the base station downlink beams could be signaled to the UE together with a map of base station positions to perform the positioning at the UE, the Angle of Arrival (AoA) of the base station uplink beams could be utilized to perform the task on the network side.

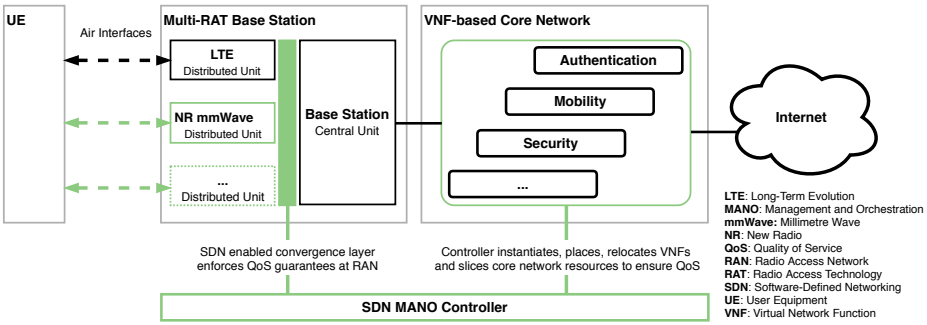


Fig. 5.56: End-to-end system architecture design for aggregated Radio Access Technologies (RATs) with prioritized services. The slices with different service requirements are orchestrated by the Software-Defined Networking (SDN) Management and Orchestration (MANO) controller. ©[2019] IEEE. Reprinted, with permission, from [266].

With the former providing reportedly sensitive information of base station positions to the UE, the latter alternative requires an active transmission of the UE, but no additional utilization of radio resources/signaling overhead leading to a resource-efficient positioning solution. In both cases, a preferably accurate direction estimate is required for positioning, which might be feasible only in case of a high-resolution sampling of the angular exploration space. Additionally, a Dilution of Precision (DOP) can be observed at acute angles between the intersecting bearings, so an elaborate placement of base station antennas might be advantageous.

5.5.4 Integration of High Priority mmWave Links into an End–to–End System Architecture

As part of 5G, the mmWave spectrum contributes to the available resources at the Radio Access Network (RAN). End-to-end applications between users (humans or machines) and services come with various requirements, which differ greatly from each other. Within an end-to-end system architecture, a mobile network needs to be agile and utilize the available resources at both, the core network and the RAN so that the application requirements can be fulfilled. As already elaborated in Section 5.4, network slicing is introduced to define virtual networks with certain configurations regarding throughput, latency, reliability, and others. Based on this, each application is dedicated to a specific *slice* that not only supplies the required performance, but also remains unaffected by traffic fluctuations or shortcomings of other slices of the same mobile network. To illustrate the potential of this slicing, our work [266] presents a system concept and an experimental evaluation of an unaffected and prioritized communication link among other *best-effort* traffic.

The overall system architecture design is depicted in Figure 5.56. To ensure the Quality of Service (QoS) requirements, the proposed Software–Defined Networking (SDN) Management and Network Orchestration (MANO) controller affects both RAN and core network. At the RAN, multi-RAT base stations are capable of directing the data traffic flow to and from UEs through different air interfaces according to the guidelines from the SDN MANO controller. For example, a conventional LTE link can be used in parallel with a 5G New Radio (NR) link at the mmWave domain at different Distributed Units (DUs) of the same base station. At the same time, the base station Central Unit (CU) is connected to the core network, where Virtual Network Functions (VNFs) dynamically allocate resources as required to operate the appropriate services. Finally, this design ensures end-to-end QoS within the whole mobile network. An experimental proof of concept study can be found in [266], where the Software-Defined Radio (SDR) and SDN-based components of the experimental setup allow for high flexibility and adaptability.

5.5.5 Intelligent Reflectors for Enhanced Propagation and Coverage under Non-Line-of-Sight Conditions

In addition to the discussed propagation loss and the need for directional transmissions, mmWave signals barely penetrate materials. As a consequence, the outdoor-to-indoor coverage is rather poor and obstructed areas need to rely on the presence of suited reflection paths. These reflection paths in turn are volatile and need to be explored by means of a potentially time-consuming discovery procedure such as the aforementioned *beam sweeping*. The beam management needs to provide routines to recover from link blockages and to switch between propagation paths, whenever the LOS condition varies. Taking the NLOS propagation into account, several challenges arise regarding the mobility support, which is doubtless a crucial feature of mobile radio networks. However, especially in dense urban scenarios, frequent LOS obstructions may demand sophisticated procedures to facilitate radio links via reflection paths.

With the novel concept of *smart radio environments* and the Reconfigurable Intelligent Surface (RIS) technology, the radio channel itself becomes modifiable to enhance the transmission performance. While much research concentrates on the optimization of transmitter and receiver techniques, the idea of this concept is to deploy elements (surfaces) with controllable reflection characteristics in the environment. Hence, these RISs act as dynamically controllable passive reflectors. In this way, they enable the purposeful utilization and adjustment of reflection paths allowing for an enhanced tracking capability of user devices with an obstructed line-of-sight to the base station. [46]

In our work [267], we highlight the potential of RISs for an enhanced mmWave network coverage at an urban campus scenario, as illustrated in Figure 5.57. The simulation model is based on our previous work [631] and extended to also account for RIS.

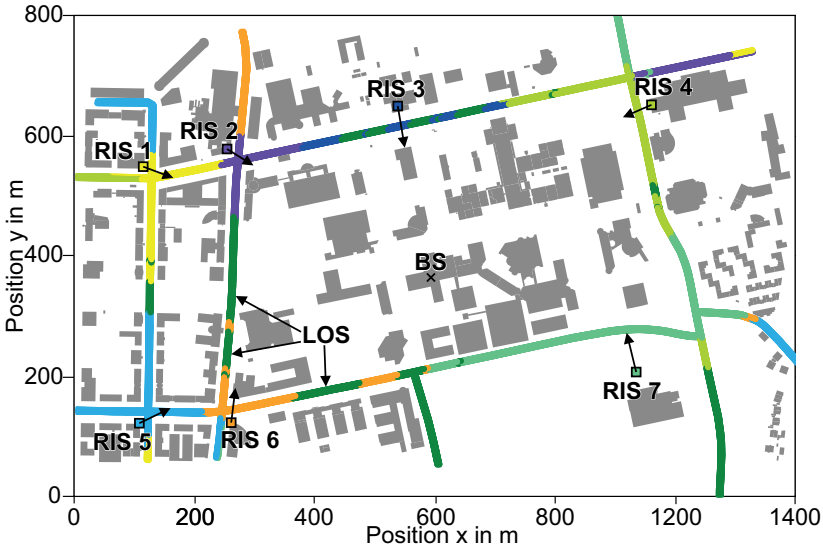


Fig. 5.57: Simulation scenario for a RIS-enhanced coverage study. With LOS coverage marked in dark green, colors according to the RISs are used for the road sections with NLOS conditions, which are covered by the respective RIS reflected paths. ©[2020] IEEE. Reprinted, with permission, from [267].

As depicted in Figure 5.58, the base station deployment only leads to a poor LOS coverage. The corresponding path loss lies only 23 % within the expected link budget of 142 dBm. However, the utilization of RIS reflection paths enhances the overall network coverage. For the alignment of the RIS reflection, distorted information about the target location is assumed. The true UE position is superimposed by a zero–mean normal distribution with standard deviation σ. The Empirical Cumulative Distribution Function (ECDF) of the path loss illustrates, that a coverage of 91 % is achievable in the case of $\sigma = 3$ m, for example. Even a comprehensive coverage is feasible due to the RIS placement in this evaluated scenario.

As a result, the deployment of RISs for smart radio environments may not only lead to an enhanced network coverage; it also allows for an improved efficiency in terms of energy and spectral resources, since the controlled reflections may reduce the exploration overhead of beam management algorithms as well as the required transmit power for sufficient signal strengths at the receiver. Nevertheless, an elaborate control of the RISs may require some radio resources for both measurements and signal suitable reflection paths. Finally, the RIS placement task may play a part in the network planning procedure and could be realized by means of machine learning approaches focusing on a cost–efficient way of providing a comprehensive network coverage.

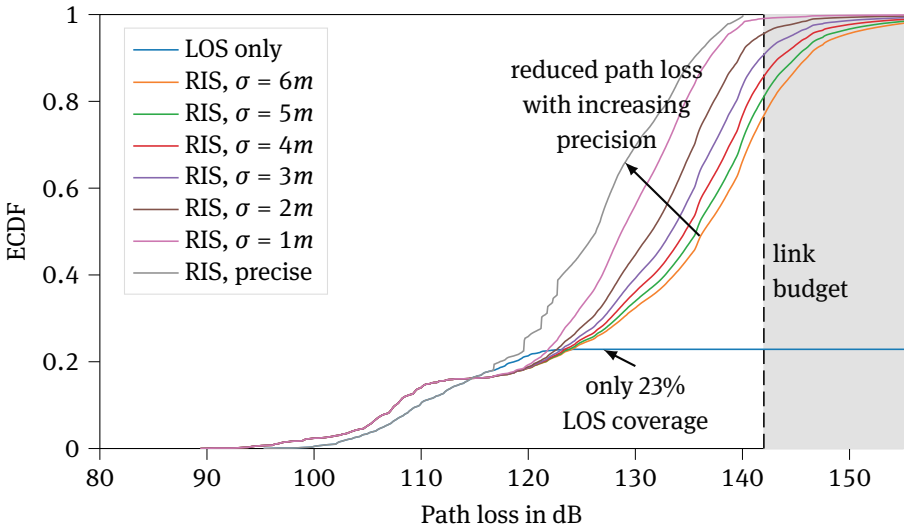


Fig. 5.58: Simulation results of RIS-enabled communication via reflection paths. Even with a slightly distorted target position information (zero-mean normal distribution with standard deviation σ), a comprehensive coverage of 91 % is achievable for $\sigma = 3$ m, while the pure LOS coverage amounts to only 23 % in this scenario. ©[2020] IEEE. Reprinted, with permission, from [267].

5.5.6 Towards Perceptive mmWave Networks by Channel Sensing

Over the past decade, it has been shown that radar and communication functionalities may be provided by Joint Communication and Radar/Radio Sensing (JCAS) systems, because the employed OFDM waveform of current 4G/5G networks and WLANs is also suitable for radar services [657]. Nowadays, a deep integration of radar-like sensing services is expected for 6G, thus allowing communication networks and their entities to become perceptive of the immediate surroundings. Such information may then be used to optimize the network performance by, say, supplying mmWave beam sweeping and tracking algorithms with user position and mobility information. Such information may also be used in the sub-6 GHz band, e.g. to assist handover decision making.

Radar systems are capable of detecting targets by analyzing the reflected waves of its own transmit signals over time. Through the use of large bandwidths and sweeping of highly directional antennas, it is possible to estimate distance, velocity, and angle information of the detected targets with high accuracy. Typical radar systems operate at very high frequencies, for example mmWaves or beyond, where multipath-based distortions are mitigated such that the high resolution due to bandwidth and directionality comes to fruition. Moreover, the imaging of the surroundings is also possible with radar technology. Considering the particular compatibility between radar requirements and mmWave communications, this is an opportunity for network operators to offer new

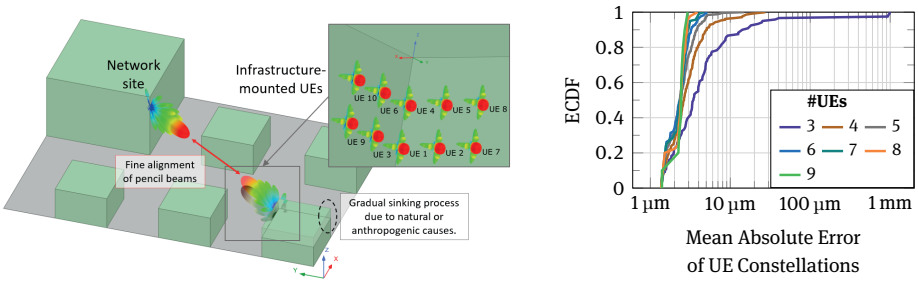


Fig. 5.59: *(a)* Scalability analysis of a subsidence process (gradual sinking of up to 5 mm) affecting a suburban house. *(b)* Distribution of mean incurred error throughout Δz-traversal for various UE mounting setups. ©[2021] IEEE. Reprinted, with permission, from [253].

services to the public, such as sensing-assisted traffic, but it also offers the prospect of process optimization in industrial facilities employing private network solutions.

However, the integration of radar functionality into mmWave communications still has a long way to go. For example, there is a need for hardware and signal processing enhancements. Nonetheless, radio-based sensing features such as user positioning have been available for more than two decades and steadily been enhanced ever since. Such sensing is enabled by analyzing the properties of one or more channels between the network and the user equipment entities. A large number of channel-based services, such as vehicle detection and classification (cf. Section 4.2), have already been proposed in literature [753]. These have been designed predominantly for sub-6 GHz WLANs, yet led to the recent launch of IEEE 802.11bf *Wi-Fi Sensing* standardization which even pertains to the mmWave domain.

With the inclusion of mmWave frequencies into the 5G standard, 5G positioning was successively adapted to allow the facilitation of mmWave beam information which, for example, enables angle-based positioning, thus enhancing the network's location services by new methods (cf. Section 5.5.3). Our work [253] followed a similar approach and considered the use of pencil beam orientation information to enhance traditional channel phase tracking-based measurements of relative motion and vibrations. By combining the movement information of several UEs along the LOS path beam orientations, we showed that millimeter range motions may be reconstructed in 3D space with less than 10 µm error. (See Figure 5.59 for the detailed results of a sample scenario.) Our full scalability analysis suggests that the usage of 4 to 5 distinct spatial link opportunities, which are expected in typical urban deployments between a single TX-RX pair, is a sensible choice. Therefore, high accuracy 3D motion tracking could in the future be conducted with a single-user device exploiting several distinct propagation paths to the network. Our ongoing work is evaluating the achievable beam orientation accuracy and the consequences of misalignment in the prior two contexts. Nonetheless, by showing

that future sensing features may already be realized with current mmWave technology, we point out the need for more research on this area because such techniques may also allow mobile networks to become more perceptive of its surroundings.

5.5.7 Concluding Remarks

In this section, the potential of mmWave communication has been elaborated with insights into several promising areas of research. The novel mmWave spectrum for mobile networks embodies a great opportunity for various future applications due to the vast amount of available spectral resources as well as the peculiar radio channel conditions. The demand for directional communication offers less interference and better spatial reuse of the spectral and time resources. According to the results of our presented works, the challenge of a proper beam alignment appears manageable and beam-based positioning can be provided as an additional feature. Also, there are concepts for integrating the novel spectrum into the overall mobile network capacity in terms of network slicing as shown in terms of a systems perspective. Last, the field of application will be further enhanced by the introduction of the novel concept of smart radio environments, where RISs support the propagation of mmWave beams and thus enhance the comprehensive network coverage in obstructed areas. Via an outlook on future perceptive networks, we showed that current networks could already partially enable novel sensing applications as expected for 6G. Therefore, research should further investigate and test sensing techniques that employ mmWave beam orientation information alongside the ongoing development of a 6G JCAS framework.

5.5.8 Acknowledgments

In addition to the CRC 876, part of this work has been partially supported by the Ministry of Economic Affairs, Innovation, Digitalization and Energy of the State of North Rhine-Westphalia (MWIDE NRW) along with the Competence Center 5G.NRW under grant number 005-01903-0047.

6 Privacy

6.1 Keynote: Construction of Inference-Proof Agent Interactions

Joachim Biskup

Abstract: To comply with the social issue of preserving privacy or pursuing other confidentiality requirements, we outline a broad range of conceptual solutions to a task of computing engineering: configuring the formal interactions of an individual's information system agent with the client agent of a communication partner in an inference-proof manner. Here inference-proofness means the following. A security mechanism shielding the system agent under the individual's control is reducing the information content of the messages sent to the client agent such that the partner would not be able to learn any information to be kept confidential under the individual's confidentiality concerns. This goal has to be provably guaranteed even if the communication partner as a rational reasoner will exploit not only a priori knowledge about the application underlying the communication acts but also additional background knowledge comprising both a complete specification of the interaction semantics and the full awareness of the security mechanism.

6.1.1 Foreword: Intended Audience

This contribution gathers, unifies, clarifies, and explains in depth the concepts and insights of a dedicated line of research and development within one of the basis subfields of IT-security, namely user-centric, self-determinative, and computer-supported enforcement of confidentiality interests, including the preservation of privacy at the discretion of the individuals involved. The own contributions started around two decades ago (see the brief bibliographic notes in Section 6.1.7), at the beginning inspecting the evaluation of sequences of closed queries by a database management system, such as a relational database system, the abstract semantics of which are based on a fragment of first-order logic. Understanding IT-security as a comprehensive problem of both organizational and computational issues, over the time, it becomes more and more demanding to expand to further operations like, e.g., transaction management for mixed query and update operations by more expressive information processing systems. At the end, dealing with procedural program execution as a service of any kind of knowledge- and belief-management system would be the ultimate goal.

The broader the range of particular operations by specific computing systems, in each case treated by appropriate highly sophisticated means, the more urgent the

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.
https://doi.org/10.1515/9783110785982-006

need to identify useful abstractions for the computational issues and to reconsider the supportive organizational issues becomes. Regarding the first point, we abstract the object of protection to be the epistemic state of an intelligent computing agent participating in a multi-agent computing configuration. Regarding the second point, we explicitly expand all the intuitive assumptions underlying the various computational protection efforts into a framework of eight basic features, to which then the formal notion of the kind of protection we want to mathematically verify refers.

Experts in the fundamentals of IT-security might benefit from this article by learning the carefully elaborated essence of a large number of highly specialized publications. Computer scientists with a broader expertise in the field of confidentiality enforcement might be encouraged to generate a similar retrospective of their own line of research and, maybe, to fill some of the many gaps in the list of operations already treated. Researchers working on machine learning and embedded systems with a strong interest on security issues might also be triggered to fill those gaps for which they have appropriate expertise. Other researchers working on machine learning and embedded systems might gain detailed exemplary insights into the subtleties of integrating purely functional aspects with concise security considerations. They might further reflect on the related notions of (syntactic) data on the one hand and inferred (semantic) knowledge and belief on the other hand underlying their own work, and they might consider the design of an overall system architecture of their interest where the security measurements are appropriately located. Finally, admitting that this article deals only with a possibilistic version of confidentiality, all kinds of readers might think of and contribute to generalize the entire approach to probabilistic considerations.

6.1.2 Confidentiality-Preservation and Inference-Proofness

Since time immemorial, among many other activities, and in a closely intertwined manner, people have reasoned as individuals by acquiring, structuring, keeping, and exploiting *information* to make up their respective minds and behaved as social creatures by *communicating* with others. With the advent of computing technologies, individually dealing with information and socially communicating have been partly delegated to *computing agents*. On the one hand, the delegation is meant to facilitate routine tasks or even enhance human capabilities.

On the other, depending on the context, as delegators, individuals at their discretion or groups of them according to some socially accepted norm aim to still control the computing agents executing protocols as their delegatees, or at least the human delegators should appropriately configure the computing delegatees.

Being aware of the resulting reduction, and somehow simplifying, we can map concepts of human reasoning and communication to the inference protocols and interaction protocols of their computing agents and, correspondingly, actually performed human activities to protocol-complying computing process executions. Under such a

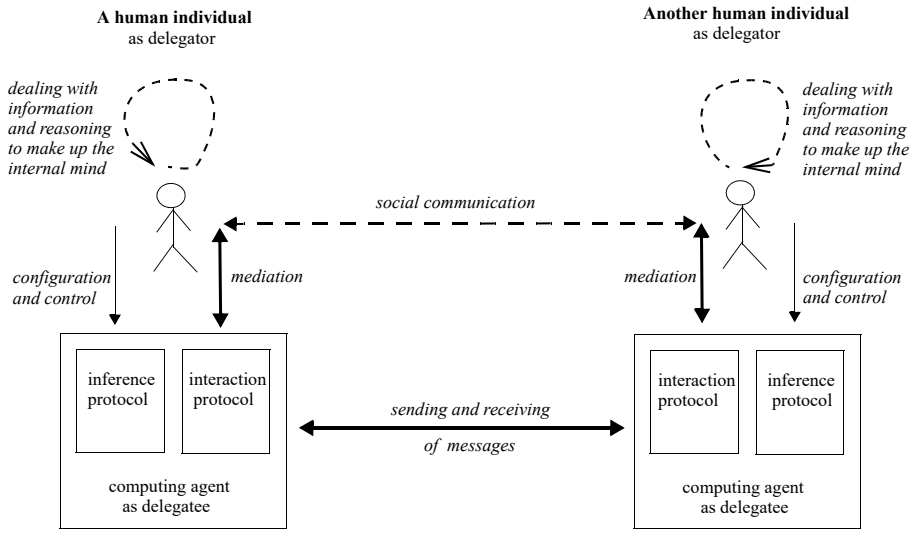


Fig. 6.1: Two reasoning and socially communicating human individuals and their protocol-based interacting computing agents as part of a larger community and the corresponding multi-agent computing configuration.

reduction, and even more simplifying, a group of human individuals is modeled to be complemented by a *multi-agent* computing configuration. In this model, each *human individual* controls a dedicated computing agent that, at least partly and by means of *protocol executions*, both deals with the information owned by that individual, in particular by internally deriving an epistemic state from a chosen information representation, and mediates the communications of that individual, in particular by sending and receiving *messages* according to one or more agreed interaction protocols. Figure 6.1 illustrates the sketched scenario.

Though, in principle, each individual can act in diverse roles and, correspondingly, each controlled computing agent can execute diverse protocols, we further specialize the model sketched above in focusing on only two individuals together with the respective computing agents. One individual is seen as an *information owner* controlling an *information system agent*, and the other individual is treated as a cooperating *communication partner* employing a *client agent*. Moreover, to enable *cooperation*, in principle the information owner is willing to *share information* with the communication partner. However, complying with *privacy issues* or pursuing other *confidentiality requirements*, as an exception from sharing, the information owner might want to hide some specific *pieces of information*.

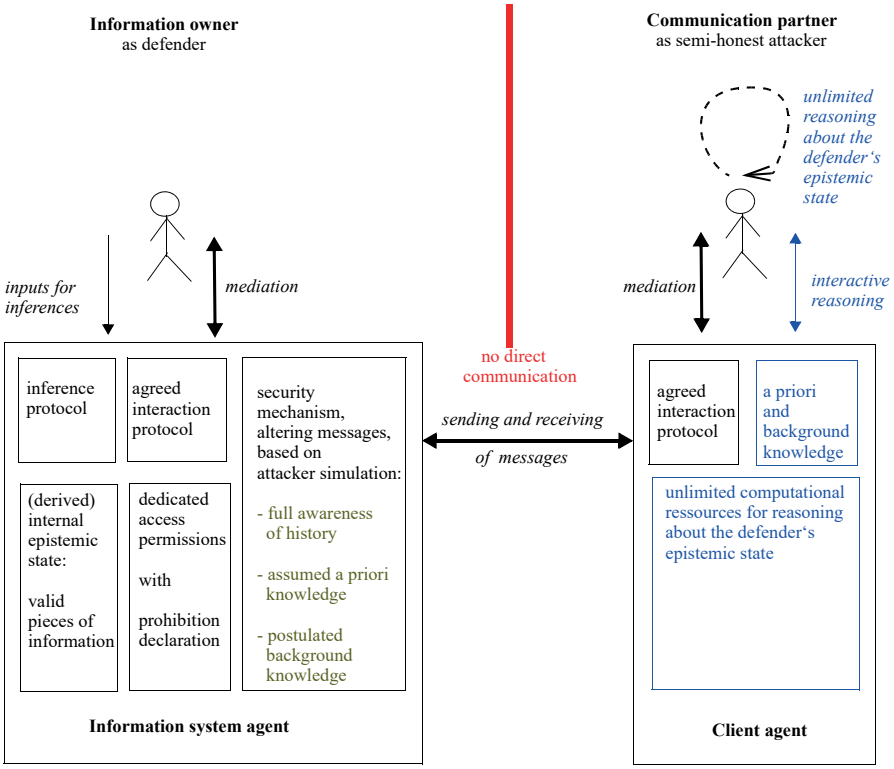


Fig. 6.2: The framework of a defending information owner with his information system agent and an attacking communication partner with his client agent (showing suspected but inaccessible parts in blue). See Figure 1 on page 82 of [73], © IFIP/Springer 2020.

Slightly more concisely, and visualized in Figure 6.2, we assume the following *framework* with eight *features*.

1. [Epistemic state of the information system agent as a single object of protection.] The human information owner does not deal with information processing and reasoning by himself but only provides the inputs to the information system agent under his *control*. At each point in time, that agent is internally deriving a formally defined *epistemic state*.
2. [Mediation of human communications by interacting computing agents.] Once having agreed on cooperation, the human information owner and his human communication partner do not communicate directly with each other, but only mediated by the computing agents under their respective control.
3. [Dedicated access permissions for information sharing.] Independently of the actual epistemic state, the information owner has granted dedicated *access permissions* to his communication partner. These permissions

declare that over the time the client agent of the partner may *interact* with the information system agent of the owner following some explicitly chosen *interaction protocols* that exclusively refer to the internal epistemic state of the information system agent.

4. [Exceptions by explicit prohibitions designating pieces of information.]
   Also independently of the actual epistemic state, the information owner has explicitly *declared* exceptions from the dedicated access permissions in the form of prohibitions. Each *prohibition* specifies a piece of information that the communication partner should not be able to learn. More precisely, with each prohibition expressed in terms of the information system agent and thus in reference to possible epistemic states, the communication partner should never be able to become sure about the actual *validity* in the epistemic state of the information system agent. In other words, from the partner's point of view it should always appear to be *possible* that the prohibited piece of information is *not valid* in the epistemic state of the information system.
5. [Partner suspected to reason about the validity of prohibitions.]
   Though the client agent is restricted to follow the interaction protocols of the access permissions exactly, the human communication partner can choose any sequence of permitted commands. Moreover, the partner is assumed to have unlimited computational resources when rationally reasoning about the validity or non-validity of a prohibited piece of information.
6. [Security mechanism implanted in the owner's information system agent.]
   To actually enforce the confidentiality requirements of the information owner, the information system agent is enhanced by some implanted *security mechanism* that should shield the underlying information processing from direct contact with the client agent. That security mechanism first inspects each message to be sent by the information system agent to the client agent according to the pertinent interaction protocol whether a *violation* of the information owner's confidentiality requirements would be enabled on the side of the communication partner. If this is the case, the security mechanism then *alters* the message such that the message is still as informative as possible but all options for a violation are blocked.
7. [Reasoning supported by a priori knowledge and background knowledge.]
   First of all, the communication partner's rational reasoning about the internal epistemic state of the information system agent is based on the messages exchanged by the respective computing agents. These messages are completely known to both agents. Additionally, the partner's rational reasoning is presumed to be supported by some *a priori knowledge* about the application dealt with in the cooperation between the two individuals involved and additional *background knowledge* comprising both a complete specification of the *interaction semantics* and the full *awareness of the security mechanism* (possibly even including the prohibition declaration) and, most notably, nothing else.

8. [Principle inaccessibility of the partner.]
   The internals of both the human communication partner and his client agent are considered to be principally inaccessible for the information owner and his system agent. This implies that the latter ones can only rely on *assumptions* about the details of the a priori knowledge and a *postulation* about the background knowledge available to the former ones.

These features can still be formally instantiated in various ways. In all cases, we follow a martial-sounding but common terminology, which ignores that in many scenarios an individual involved as communication partner will primarily be treated as cooperating in a friendly manner, rather than as a "total enemy". At least partially trusted for consciously sharing information in principle and correctly executing the agreed interaction protocols, the *communication partner*—together with the *client agent* controlled by him—is denoted as a *semi-honest attacker*, suspected to potentially aiming to maliciously infer the actual validity of pieces of information that the information owner has declared to be kept confidential. Accordingly, the *information owner*—together with the *information system agent* controlled by him—is denoted as the *defender*.

Now, the security mechanism has to invariantly enforce a suitable version of the following *security policy* of (possibilistic) *inference-proofness*, which also specifies the *attacker model*: For each prohibited piece of information ψ, the information content of messages sent to the attacking client agent—which is possibly enhanced by reasoning capabilities supplied by the human communication partner—during executions of agreed interaction protocols will never enable the attacking receiver to rationally infer that ψ is valid in the epistemic state, even when

- inspecting the complete *history* of preceding *interactions*,
- considering some *a priori knowledge* about the possible *epistemic states*,
- applying the *semantics* of the agreed *interaction protocols*, and
- being *aware* of the *functionality* of the *security mechanism*.

The concept of *rationality* on the side of the attacker is then captured by the following rephrasing of the still to be suitably versioned security policy of *inference-proofness* in terms of *indistinguishability* (as roughly visualized by Figure 6.3):

For each prohibited piece of information ψ,
for each epistemic state d satisfying the a priori knowledge,
for each sequence of messages $mes_1, \ldots, mes_k$
  exchanged during an interaction history
  complying with the agreed interaction protocols
  but potentially altered by the security mechanism
there exists an "*alternative*" epistemic state d' such that

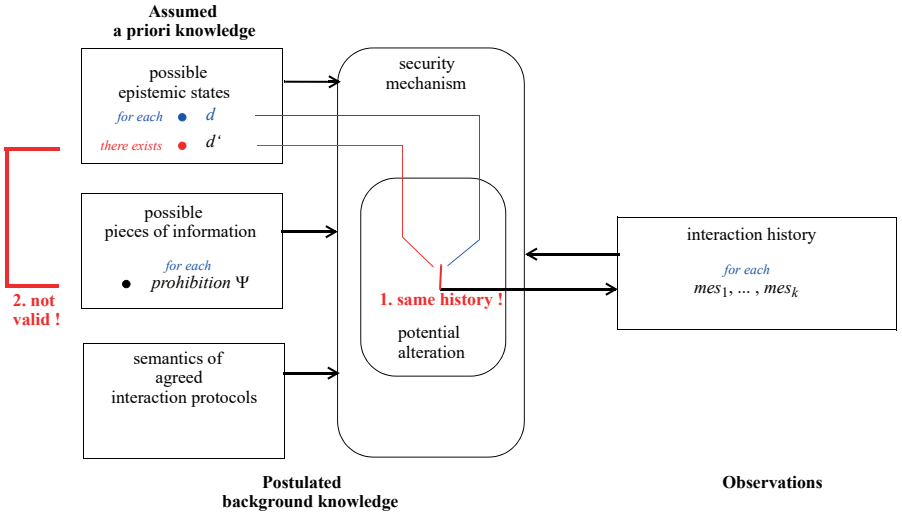


Fig. 6.3: A rough visualization of the requirement of inference-proofness: "for each ψ, for each d, for each $mes_1, \ldots, mes_k$, there exists d' such that ...".

1. the same sequence of messages would be generated, in compliance with the agreed interaction protocols and subjected to the alterations by the security mechanism, but
2. ψ is not valid in d'.

For this rephrasing, the epistemic state d is thought as actually derived by the information system agent—for short, "stored"—and might satisfy the prohibited piece of information ψ or not. The former case implies that the alternative state d' required to exist is different from d; in the latter case, the actually stored state d and the alternative state d' might be the same. Accordingly, declaring ψ as a prohibition does not intend to block any option that enables the attacker to infer the non-validity of ψ.

Confidentiality as inference-proofness could be trivially achieved by granting no access permissions at all or altering the information content of all messages sent to the attacker to nothing, violating any conflicting *availability requirements* and shutting down any communication mediated by the respective computing agents and, thus, making the whole thing useless. Accordingly, confidentiality requirements and availability requirements always have to be suitably balanced.

All our work focuses on the following three-level *conflict resolution* strategy:

1. As a general rule, some dedicated access permissions are granted for the sake of *availability*, to be freely enjoyed by the client agent, insofar as they do not conflict with level 2 of the strategy.

2. As exceptions from the general rule, specific prohibitions are declared for the sake of *confidentiality* that never must be violated and, thus, these prohibitions have to be enforced by alterations made by the security mechanism, but to comply with level 3 of the strategy only insofar as definitely necessary.
3. As a limitation for the effect of exceptions, the alterations made have to be minimal, again for the sake of *availability*.

Given the access permissions on the first level, the second and the third level lead to a combination of a *constraint solving problem* and an *optimization problem*.

Our simplified defender-attacker agent model still allows many instantiations and versions, respectively. For our concrete ongoing research, and accordingly for this article, we grossly distinguish the *structures* of three fundamental *data types* for an *information system*, namely *abstract* data sources, *propositional* knowledge or belief bases, and *first-order relational* databases.

For each of these data types—considering suitable refinements—we deal with the pertinent *operations*, which in our case are the *interactions* with a client agent, comprising in all cases at least

– *closed-query /yes–no-query evaluation* with *response* preparation,
  performed repeatedly with queries that in general are different;

and, depending on the refinement, additionally

– *open-query evaluation* with *response set* preparation,
  performed repeatedly with in general different queries;
– *view generation*,
  performed only once, since the attacker can freely employ a received view at his discretion instead of contacting the defender again;
– *view updating*,
  possibly performed from time to time, if manageable at all;
– *knowledge update transaction*,
  performed repeatedly usually intertwined with queries;
– *belief revision*,
  performed repeatedly usually intertwined with queries evaluated under *non-monotonic reasoning*;
– procedural *program execution*,
  performed repeatedly with in general different input parameters;
– *data outsourcing*,
  performed only once.

Tailored to the respective refinement, we propose and study *alterations* to the available interactions to ensure inference-proofness. There are two basic approaches to alterations, namely *weakening* the pertinent information about the actual epistemic state

and *lying* about the pertinent information about the actual epistemic state, allowing or even requiring appropriate refinements and *combinations* of weakening and lying.

6.1.3 A Generic Construction Methodology for Alterations

We can design and verify basic kinds of alterations according to the following generic construction methodology, which refers to the eight features of the framework described in Section 6.1.2. According to Feature 1, at each point in time, the defending information system agent is privately deriving its *actual* epistemic state. According to Feature 3, this state is then taken as the basis for the data contained in the messages to be sent to the attacking client agent during an interaction complying with some agreed protocol, for which some dedicated access permissions are granted. According to Feature 5, the attacker is suspected to aim at gaining as much information as possible about the defender's actual epistemic state, in particular whether a prohibited piece of information is valid in that state. In general, however, the attacker will face some uncertainty about that state, since (i) by Feature 2, there are no direct communication acts between the human individuals involved and, (ii) by Feature 6, the attacker has been separated from the defender's underlying information processing by the shield of the implanted security mechanism and the interactions are restricted to the exchange of messages. According to Feature 4, the attacker's uncertainty should always include that any prohibited piece of information might be *not* valid in the defender's actual epistemic state.

Conceptually, the attacker's uncertainty can be captured by the set of those epistemic states that appear to be *possible* to him. According to Feature 7, an epistemic state qualifies to be possible if it is compatible with both the potentially altered messages observed so far and the already initially available a priori knowledge and background knowledge. All the qualifying epistemic states together form the *least uncertainty* left to the attacker, i.e., the *best* achievement to satisfy his suspected curiosity:

- exactly one of the qualifying epistemic states is the actual state;
- all other qualifying epistemic states could possibly be the actual state as well;
- all non-qualifying epistemic states can definitely be excluded from being the actual one.

At the point of time t, we call the set $bestcv_t$ of the then-qualifying epistemic states the attacker's *best current view* (on the defender's actual epistemic state). In doing so we do not care whether or not the attacker really achieves exactly this optimal result. However, on the one hand, the kind of protection wanted by us is strongly based on the *presumed rationality* of the attacker: he definitely will never miss to identify an epistemic state as qualifying. On the other hand, he might be too lazy to exactly identify all actually non-qualifying epistemic states. In other words, the attacker might work with either the best current view $bestcv_t$ or any superset of it. Under this condition, the security

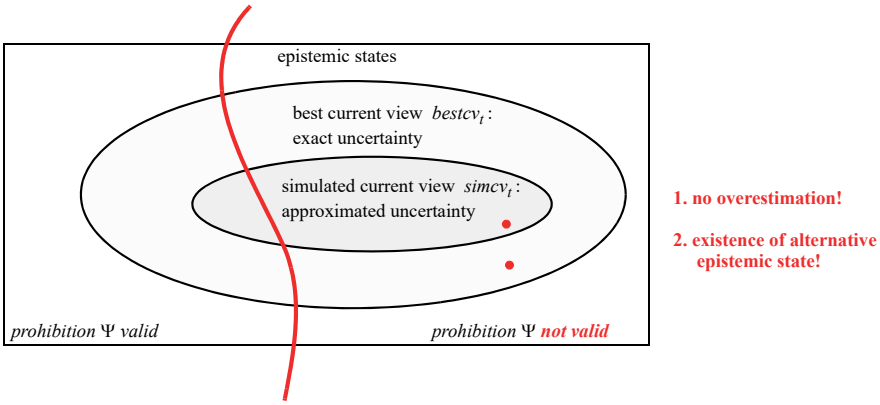


Fig. 6.4: A rough visualization of a pertinent security invariant.

policy of inference-proofness essentially requires that the best current view $bestcv_t$ still reflects *sufficient uncertainty*, namely that for all prohibited pieces of information ψ there exists an epistemic state $d' \in bestcv_t$ such that ψ is not valid in d'.
Fortunately, the defender's security mechanism does not need to determine the attacker's best current view $bestcv_t$, and the mechanism would not be able to carry out such a determination, due to the *inaccessibility* of the partner's internals according to Feature 8. Instead, it suffices to maintain an appropriate *simulated current view* $simcv_t$ that approximates the inaccessible behavior of the attacker and to enforce the following *security invariant* for all points in time t, or sometimes even a stronger one (as roughly visualized by Figure 6.4):

- $simcv_t \subseteq bestcv_t$, i.e., the defender approximates the attacker's uncertainty from below, potentially underestimating but never overestimating the uncertainty;
- for all prohibited pieces of information ψ there still exists an epistemic state $d' \in simcv_t$ such that ψ is not valid in d'.

Moreover, in general the security mechanism will work with some concise (i.e., algorithmically treatable) *representation* $rep(simcv)$ and algorithmically check the security invariant in terms of the pertinent representation.

Now, such a simulation would then be *initialized* at the point in time $t = 0$ by setting $rep(simcv_0)$ according to a suitably concise representation of the (assumed) a priori knowledge. This requires the natural *security precondition* that the a priori knowledge does not violate any prohibition. In fact we cannot prevent the attacker from "learning" what he is already sure of.

Inductively, at the point in time $t + 1$, the simulation should contain a suitably concise representation $rep(simcv_t)$ of an appropriate set $simcv_t$ of epistemic states. Then, for the functionally correct messages to be sent to the client agent—and in some cases

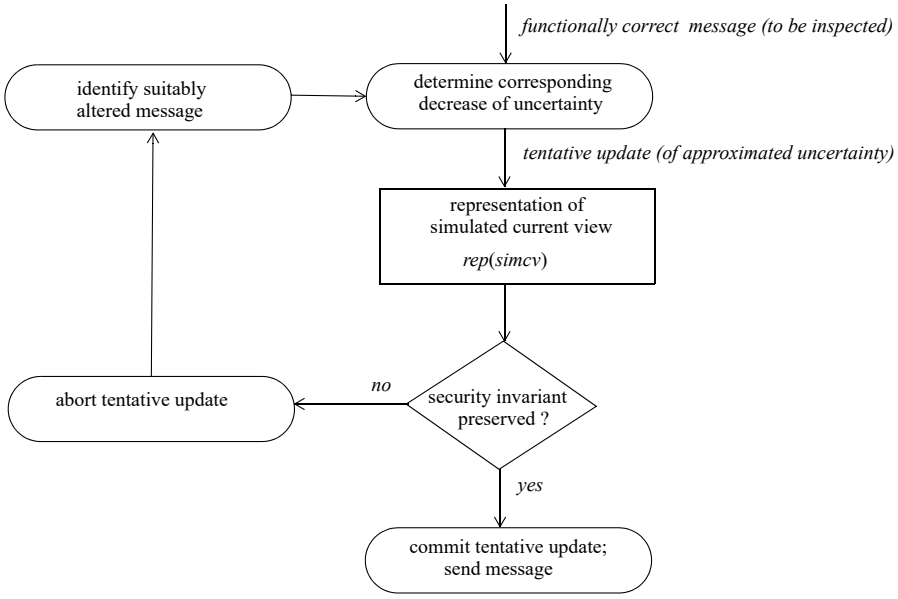


Fig. 6.5: A rough visualization of the basic steps of inductively enforcing a security invariant.

for some suitably determined modifications as well—the security mechanism first has to inspect the consequences of *tentatively* updating $rep(simcv_t)$ accordingly, i.e., whether or not the security invariant would be violated. In case of a violation, the security mechanism then has to identify suitable alterations that definitely avoid the violation. Finally, the possibly altered messages are actually sent out, and the simulation is actually updated accordingly, now ensuring the security invariant. Figure 6.5 visualizes the basic steps of these actions though the form is not universally applicable.
So we are left with the most crucial points of our construction methodology:

1. *maintaining a convenient data structure* for a "suitably concise representation $rep(simcv_t)$ of an appropriate set $simcv_t$ of epistemic states";
2. *checking tentative updates* for violations of the security invariant, preferably aiming at the approximately best computational complexity; and
3. as far as required, efficiently *identifying suitable alterations*, preferably without the need for a further explicit violation check.

Moreover, each specific situation dealt with might require some variations, in particular regarding the points in time to be considered. Let us first consider situations for which the defender's *epistemic state* is kept *fixed* over the time. If the sequence of interactions to be treated consists of only instantiations of the most basic interaction—*closed-query/yes–no-query evaluation* with response preparation—then the successive

points in time are essentially determined by the defender receiving the next query, which asks for the validity of a single piece of information in the defender's fixed epistemic state.
If the sequence of interactions also contains instantiations of the more advanced interaction of *open-query evaluation* with response-set preparation, then the response set can be formed by iteratively inspecting an internally generated sequence of closed queries, each of which is either a ground substitution of the open query (i.e., obtained by replacing free variables by constant symbols) or a specifically defined completeness sentence (dealing with "negative information"). Accordingly, for the point in time of receiving the open query, basically shared with the attacker, the defender privately determines a sequence of subpoints in time. Similarly, if the sequence of interactions also contains an instantiation of *view generation* for the fixed state, a response can be formed by internally inspecting a somehow exhaustive sequence of closed queries, also leading to a private sequence of subpoints in time. Alternatively, a view can be generated by inspecting suitably defined open queries, as explained above, leading to the respective private sequences of subpoints in time.

In general, however, the *epistemic state* of the defender might *change* over the time, raising additional issues, especially when it comes to finding an inference-proof way for

4. *resolving conflicts* between confidentiality and integrity, and
5. ensuring *backwards confidentiality*.

Basically, a confidentiality–integrity conflict might occur if the request for a change of the defender's epistemic state originates from the attacker. In general, with regard to pure functionality, the information system agent has to perform a *transaction* that first tentatively changes the state as requested and then checks whether or not the new state complies with all *semantic constraints* that are declared to be maintained as an *integrity invariant* according to the underlying application and, thus, assumed to be part of the *a priori knowledge*; for a relational database such a declaration might be part of the database schema, but in all cases such a declaration might also be expressed externally. In case of compliance, the transaction is *committed* making the tentative change persistent; otherwise, in case of non-compliance, the transaction is *aborted*, recovering the previous state. In both cases the requester is notified accordingly.

However, with regard to a security policy of inference-proofness, the respective functionally correct notification can be seen as a response set to one or more appropriately constructed closed queries. This response set might enable the attacker to learn the validity of some piece of information contained in the prohibition declaration, i.e., incorporating the response set to the (concise representation of the) *simulated current view* would violate the required *security invariant*. Consequently, the resulting conflict has to be resolved by either suitably modifying the transaction functionality or altering the response set leading to the actually returned notification, as already outlined for queries, or combining both activities.

In doing so, we also have to account for an *implicit information flow* that is caused by the overall control flow structure of a transaction as a *guarded command* in the form of an *if-then-else branching*: knowing the transaction semantics and being fully aware of the security mechanism, if the attacker can infer which branch has been selected, then he can also figure out whether and how the epistemic state has actually been changed and which alterations have actually been made to the functionally correct notification.

More generally, the defender's side of an interaction might have a more or less sophisticated overall *control flow structure* stemming from guarded commands like *if-then-else branching*, *repeat-repetitions*, *while-repetitions*, and similar procedural commands that can cause implicit information flows. Then the security mechanism typically inherits this potentially critical control flow structure. Moreover, the (code of the) security mechanism itself might have a critical control flow structure. Accordingly, the security mechanism has to appropriately treat an attacker's potential observation of a control path that has actually been chosen during a (hidden) execution on the defender's side and the resulting implicit information flow, similarly as the responses to explicit queries. As outlined above for transactions, the treatment might include to interpret such an observation as an implicit query.

The issue of *backwards confidentiality* results from the following observations about the consequences of an update of the epistemic state. First, previously released information about the validity of a prohibited piece of *information* might become *outdated* and, at least in general and whenever possible in an inference-proof way, the defender should suitably inform the attacking partner about the occurrence of an update and send him a pertinent *refreshment* of outdated information. Second, it can be shown that such a notification together with the refreshment and further information contained in messages sent at subsequent points in time might enable the attacker to infer the validity of a prohibited piece of information in the past at some preceding point in time. Accordingly, we have to strengthen the *security policy* of inference-proofness by requiring *continuous inference-proofness* to be enforced for the full range of all points in time that so far have happened, rather than just for the respective last one. This goal can actually be achieved by checking tentative updates of the representation of a simulated current view for *stronger versions* of the *security invariant*.

Basically, the general construction methodology for alterations proceeds iteratively, whether the points in time considered are externally determined by observably receiving/sending a message or only privately generated. However, for an interaction expected to be performed only once, in particular for *view generation* and *data outsourcing*, there could be only one point of time of interest, and thus one might look for a security mechanism that is working "more globally". Indeed, for view generation we have successfully designed and verified such a mechanism, in addition to the iteratively working ones inspecting sequences of appropriately formed queries, as mentioned before. Furthermore, the security mechanism for outsourcing data does not rely on such sequences. Nevertheless, these cases are also inspired by the approach to conceptually

set up some simulated counterpart to the best current view, possibly represented in some more manageable way.

6.1.4 Specific Constructions for Alterations

6.1.4.1 Weakening Including Refusing

The *weakening* approach to alterations of a harmful message comprises the special case of *refusing* to provide any *explicit information flow* as its extreme form.

In terms of a complete logic a refusal can *literally be represented* as the tautology $\chi \vee \neg\chi$, known by a rational attacker right from the beginning without having performed any interaction with the defender. However, literally replacing χ by the tautology $\chi \vee \neg\chi$ simply only if the actual validity of χ is harmful, without additionally caring about the potential harmfulness of its non-validity, equivalently by completeness, the fictitious validity of $\neg\chi$, would trigger an *implicit information flow* by means of the following kind of *meta-reasoning*, which exploits the postulated background knowledge about the security mechanism:

> For a (definitely flawed) security mechanism that, while inspecting a valid sentence χ, refuses on the harmfulness of χ but not on the harmfulness of the fictitious validity of $\neg\chi$, we would have the following equivalence: refusing occurs if and only if χ is valid and harmful. Thus, observing a refusal, the attacker could infer the validity of χ.

Notably, this kind of reasoning would just be caused by the careless handling of the critical *control flow structure* in the form of an if-then-else branching.

Accordingly, to ensure the required inference-proofness, the security mechanism has to make the two possible cases of validity and non-validity *indistinguishable* for the attacker, in the simplest way by refusing if and only if at least one of the cases is harmful.

In some situations weakening can also be achieved by using more general disjunctions. For example, let both ψ_1 and ψ_2 be prohibited pieces of information in isolation but not the disjunction $\psi_1 \vee \psi_2$, i.e., knowing the validity of $\psi_1 \vee \psi_2$ in the actual epistemic state is considered to be harmless, but figuring out which of the two disjuncts leads to the validity is harmful. Appropriately taking care of options for meta-reasoning similarly as for pure refusing and, additionally, of the potential entailments among several such disjunctions and of such disjunctions with other pieces of information, we might ensure inference-proofness by replacing a valid information ψ_i by the disjunction $\psi_1 \vee \psi_2$.

For all such kinds of weakening, the *literal representation* of weakened information, as conveyed in messages to the attacker,

- *correctly* reflects the actual epistemic state of the defender, and
- *notifies* the attacker about the fact of an actually performed weakening.

In terms of a logic, the correctness property allows us to solve the first three crucial points for the construction methodology roughly as follows:

1. A *simulated current view* is directly *represented* by the set of those sentences that so far are known to be valid in the defender's epistemic state, according to the a priori knowledge and the literal contents of messages.
2. A tentative update is independently performed with both the inspected sentence and its negation, respectively, and both versions are *checked for violations* of the security invariant by solving implication/entailment problems of the form "[current set of sentences together with tentatively added *sentence or its negation*, respectively] entails [sentence designating a *prohibition*]" by a pertinent *theorem prover*.
3. If any of the checks is positive, refusing is straightforwardly *identified* as the *suitable alteration* (or, possibly, a less easily defined but more informative disjunction that is stronger than a tautology but still harmless), and then the alteration can be notified in the corresponding message, leaving the current representing set of sentences untouched (or updating that set by adding the identified disjunction).

6.1.4.2 Lying

In contrast to weakening, the *lying* approach to alterations requires a sharp distinction between the *literal representation* of responses and the attacker's *rational sophisticated conclusions* about what he is literally observing.

To start with literal representations, consider for example a directly prohibited piece of information ψ in terms of a complete logic. The security mechanism would always have to pretend *literally* the non-validity of ψ. Hence, if the attacker is *aware of the security mechanism* including the prohibition declaration, he would not need to query any prohibition ψ: he will *always* receive the literal response that $\neg\psi$ is valid.

This feature implies that the defender cannot expect the enforcement of a declaration treating both ψ and $\neg\psi$ as prohibitions. Less obvious is a further consequence: even if the defender declares several pieces of information $\psi_1, \ldots, \psi_k$ as individual prohibitions, nevertheless under lying the disjunction $\psi_1 \vee \cdots \vee \psi_k$ has to be protected literally as well. For otherwise the attacker could perform the following *inconsistency-reasoning*:

> For a (definitely flawed) security mechanism that only lies literally on the (explicitly declared harmful) validity of $\psi_1, \ldots, \psi_k$ but not on the validity of $\psi_1 \vee \cdots \vee \psi_k$, we could get a literal representation of responses that contains the following inconsistent set of sentences: $\{\neg\psi_1, \ldots, \neg\psi_k, \psi_1 \vee \cdots \vee \psi_k\}$. Thus, observing such an inconsistency, the attacker could identify the occurrence of some lying in the literal representation and, furthermore, could be tempted to infer the validity of any sentence.

Accordingly, to ensure the required inference-proofness, the security mechanism better has to avoid running into such a somehow *"hopeless" situation*.

Moreover, regarding rational sophisticated conclusions about a *possibly lied* assertion about the non-validity of a prohibition ψ and the literal representation of this

assertion as $\neg\psi$, the *best current view bestcv* would contain *pairs* of epistemic states that differ (at least) in one state making the sentence ψ valid and the other state making the negated sentence $\neg\psi$ valid. So, evidently, a concise representation *rep*(*simcv*) of a *simulated current view simcv* containing only a sentence $\neg\psi$ that is possibly but not necessarily a *literal lie* would represent both alternatives (as already said above, in contrast to the properties of weakening).

The two issues have a conceptually simple and provably effective solution. First, we *replace a prohibition declaration* consisting exactly of the single pieces of information $\psi_1, \ldots, \psi_k$ by the singleton containing one disjunction $\psi_1 \vee \cdots \vee \psi_k$. Such a replacement considerably strengthens the security policy of inference-proofness: for each epistemic state d under consideration, the existence of an "alternative" state d' is required such that all of the ψ_i are *simultaneously not* valid in d'. Second, we form a concise *representation* of the *simulated current view* by the literally provided responses, being aware that the same literal representations for weakening and lying, respectively, essentially differ in their semantics.

Of course, similarly as for weakening but somewhat more subtly, lying is also due for any *harmful sentence* χ that could lead the attacker to believe in the validity of the disjunction of all ψ contained in the prohibition declaration.
In summary, for all kinds of lying, the *literal representation* of lied information, as conveyed in messages to the attacker,

- might *not correctly* reflect the actual epistemic state of the defender and, thus, might seriously *mislead* a naive receiver, and,
- naturally, *does not notify* the attacking receiver about the fact of an actually performed lying and, thus, lays the burden on the receiver of finding out whether or not some lying has *potentially* occurred.

Moreover, in terms of a logic, the first three crucial points for the construction methodology are solved roughly as follows;

1. A *simulated current view* is indirectly *represented* by the set of those sentences that so far have been *pretended* literally to be valid in the defender's epistemic state, according to the a priori knowledge and the content of messages.
2. A tentative update is *checked for violations* of the security invariant by solving one implication/entailment problem in the form "[current set of sentences together with tentatively added sentence] entails [sentence designating the *disjunction of all prohibitions*]" by a pertinent *theorem prover*.
3. If that check is positive, a lie on the tentatively added sentence is straightforwardly *identified* as the *suitable alteration* and then sent (without notification, of course) in the corresponding message, and that lie is inserted into the current representing set of sentences.

6.1.4.3 Combined Approaches

As we have seen, alterations by refusing on the one hand and by lying on the other are precipitated by different causes. Actually refusing is needed if *the validity or the non-validity* of the sentence inspected would be harmful regarding a *single* prohibition. Actually lying is needed if the *validity* of the inspected sentence would be harmful regarding the *disjunction* over the prohibition declaration. We might wonder whether and how we can do better by avoiding to always check the impact of both the validity and the non-validity of the sentence inspected and by never considering the strengthened security policy to protect the disjunction. This goal can be achieved by a suitable *combination of refusing and lying*:

- if the validity of the sentence inspected is not harmful regarding any prohibition, then return a message without any alteration;
- if the validity of the sentence inspected is harmful regarding some prohibition *and* also the non-validity of the sentence inspected is harmful regarding some prohibition, then return a message suitably indicating a refusal;
- if only the non-validity of the sentence inspected is harmless, then return a message suitably altered by the lie that literally pretends the non-validity.

Moreover, we can still *represent* a *simulated current view* by the set of those sentences that so far have been *pretended* literally to be valid.

6.1.4.4 Weakening versus Lying

At first glance, alterations by *lying* appear to be rather problematic, both from an ethical point of view and regarding the desired functionality. However, while lying is *ethically banned* in general, we all know of widely *accepted exceptions*, e.g., a white lie in a most critical situation, a small insincerity to avoid some larger offense, or an untruthful answer to an illegal request.

Moreover, in many cases *literally lying* might also functionally *disturb* the agreed cooperation between the communication partners, insofar as the receiver is behaving naively and "believing the lies" without further own reasoning. Even then, however, if an actually occurring alteration by literally lying does not affect an implicit or explicit *availability declaration*, we might argue that the respective interaction is at least beyond the agreement or even misusing it.

Finally, handling an interaction that is intended to change the defender's epistemic state by means of a transaction, we might face a confidentiality–integrity conflict for which an application of literal lying appears to be most natural, at least if the ethical and functional concerns are appropriately dealt with.

Above, we already distinguished between "lying" and "literal lying", and a deeper inspection of the issues indicates that such a distinction is crucial for a more information-theoretical discussion. To start with, for any kind of alteration, a sophisticated attacker can always distinguish between the conceptual notion of the *best current view*

and the pragmatically introduced notion of the *simulated current view* and its concise representation.

The *best current view bestcv* can be seen as a kind of an *inverse image*, as derived from the observed message history $mes_1, \ldots, mes_k$ and denoted by $bestcv = con_mess^{-1}(mes_1, \ldots, mes_k)$. Here, we think of the message history being composed of k' many request messages $mes_{a_1}, \ldots, mes_{a_{k'}}$ received from the attacker, and k'' many reaction messages $mes_{d_1}, \ldots, mes_{d_{k''}}$ returned to the attacker, with $k = k' + k''$, where the latter were produced for the actual epistemic state by the defender's *message-generating* function possibly applying alterations, called *con_mess*.

Exploiting the postulated background knowledge, the attacker can thus determine *bestcv* in mathematical terms, and insofar as *con_mess* is an effectively computable function, i.e., its graph

$$\{\, (es, hist) \mid es \text{ is epistemic state,}$$
$$hist \text{ composed of received } hist_a \text{ and returned } hist_d \text{ is message history,}$$
$$con_mess(hist_a, es) = hist_d \}$$

is recursively enumerable, the graph of the inverse function con_mess^{-1} is recursively enumerable as well and, thus, for each message history *hist* the inverse image $con_mess^{-1}(hist)$ is also a recursively enumerable set. Notably, under this perspective, there is no conceptual difference between refusing and lying, or between any other approach to alterations.

The concise *representation rep*(*simcv*) of the *simulated current view simcv* is a technical means employed by the defender's security mechanism to effectively—and hopefully also efficiently—enforce the pertinent security invariant. However, exploiting the postulated background knowledge, the attacker can determine *rep*(*simcv*) as well. In fact, as outlined above, for weakening and lying that representation is just formed by the literal messages sent to the attacker.

Hence, in information-theoretical terms, the main difference between weakening (with the special case of refusing) on the one hand and any method that at least sometimes literally lies on the other hand can be described as follows:

- under *weakening* the representation of the simulated current view also represents the best current view in a *straightforward* way,
- whereas under *lying* the burden of the attacker to determine the best current view might be *much harder*.

In other words, for a sophisticated and rationally reasoning attacker, (i) there are no "real lies" but only literally ones, and (ii) lying causes an essential difference between what he can literally observe and what he can conceptually conclude, and in general imposes a high computational complexity on algorithmically determining the best current view as the least uncertainty.

A more detailed comparison of refusing and lying for a sequence of queries considers the longest prefix for which the responses have been correct (for short, called "longest honeymoon"). In turns out that refusing and lying are in general incomparable regarding this notion.

Somehow surprisingly, however, if we require to protect the disjunction over the prohibition declaration also for refusing, the information contents supplied by the two approaches are exactly the same, i.e., the conceptual best current views are always equal or, in other words, two epistemic states are *indistinguishable* for the attacker in case of refusing if and only if they are so in case of lying. Moreover, we can show that an actual refusal occurs if and only if a *potential lie* occurs. This result shows again that the information-theoretical difference between refusing and lying consists in the computational burden of the attacker to find out the grade of reliability of a message received: for refusing the attacker is explicitly notified by the defender; for lying the attacker has to find out by himself by means of rational reasoning (or just by simulating a defender that applies refusing).

6.1.5 Managing Computational Complexity

The overall computational *complexity of inference-proof* interactions is basically determined by the normal functionality of the information system under protection on the one hand and the *overhead* caused by the security mechanism on the other. Regarding the impact of the former, as far as applicable, the pertinent *logic underlying* the query evaluation appears to be most crucial. Regarding the impact of the latter, both the consideration of the *interaction history* (including the assumed *a priori knowledge*) by maintaining a concise *representation* of the *simulated current view* and the *number* and the *kinds* of *checking tentative updates* for satisfying or violating, respectively, the *security invariant* are most important. As far as applicable, the pertinent logic once again determines the costs.

In general we can expect a rather high or even practically infeasible level of complexity, and in some cases also beyond effective computability. Moreover, besides first of all treating the *constraint-solving problem* to achieve inference-proofness for the sake of confidentiality, additionally we always aim at still providing good *availability* and thus are facing the *optimization problem* to actually perform an alteration only if strictly needed. Each of the problems alone is known to be computationally hard in general, so will be their combination.

Whether or not a high computational complexity can be afforded might depend on the additional *timing constraints* of the desired communication acts. In particular, if the (attacking) communication partner expects to be served by the (defending) information owner online in real time, only a minor delay would be acceptable. By contrast, insofar as the interaction of *view generation* is initialized by the (defending) information owner, all the computations can be done offline and thus might last as

long as several hours. Accordingly, as a general heuristic we favor the shift of suitable parts of the overall computational burden to offline precomputations. The duration does not matter the communication partner at all or is bounded by the time normally spent by an interactively communicating partner for some other activity.

Given that at the core of any security mechanism we have to *check the tentative updates* of the representation of the simulated current view in terms of an underlying logic for complying with a security invariant, we can attempt to decrease the number and to lessen the complexity of such checks by restricting to special cases of the sentences used for formally expressing a priori knowledge, queries, and prohibitions. The best case would be that it suffices to relate (the validity of) a sentence inspected to the prohibitions in a straightforward way without the need to consider a simulation and thus the interaction history at all. Intuitively, this case could arise if both queries and prohibitions refer to elementary and mutually independent pieces of information.

More generally, the following guidelines for identifying computationally efficient cases have been successful: diminish the potential mutual dependence of the considered pieces of information about the defender's epistemic state and, thus, the *redundancy* contained in that state; and syntactically restrict the sentences expressing such pieces of information such that the pertinent *logical entailment* problems are easily solvable. In fact, as a first example and referring to the best case regarding *closed-query evaluation*, checking tentative updates can be done without considering the interaction history, and the entailment problems can be reduced to simple text comparisons under the following conditions: epistemic states are represented by *relational instances* of a *relational schema* with *functional dependencies* in Object Normal Form, i.e., they are in Boyce-Codd Normal Form and satisfy the Unique Key Property, and the epistemic states contain only atomic sentences, i.e., logical representations of single tuples. Further examples dedicatedly relax these requirements such that checking tentative updates can efficiently be implemented by means of SQL, even for restricted cases of *open-query evaluation*.

We can also employ a wide range of *approximation heuristics* to decrease the computational complexity, first of all by relaxing the *availability requirements* in order to facilitate the resulting *optimization problem* to minimize alterations.

6.1.6 Conclusions: Naive Illusion or Promising Hope?

Our main motivation has been to design and mathematically verify technical solutions to the social issue of preserving privacy, or of any other justified confidentiality concerns. Clearly, in general these goals require the consideration of a large range of psychological, social, institutional, legal, information-theoretical, and mathematical features. In public discussions about privacy preservation in the Information Age, some voices even claim that achieving privacy has become illusionary. Without discussing

arguments in detail, we just state that we do not share this view, as do, we believe, the legal bodies issuing the pertinent legislation.

The framework underlying our achievements focuses on a *very narrow* aspect of enforcing *confidentiality*: how to support a human information owner in hiding the validity of dedicated pieces of information, referred to as declared prohibitions, contained in the internal epistemic state of an information system agent under the owner's control, while that agent interacts with the client agent run by a semi-honest and rationally reasoning communication partner. Thus, the overall *target of protection* is *only* the internal state of a technical device whose well-defined interface to the outside world is supposed to be configurable and mastered at the discretion of the controlling human individual.

In the extreme case, that interface can just be totally disabled such that—under reasonable assumptions—no human can observe the internal state of the information system agent at all, thus preventing the *availability* of any information. So, conceptually starting with a disabled interface, the real problem is to gradually allow a flow of information from the internals of an otherwise completely shielded computing agent, while still guaranteeing the wanted kind of confidentiality of the declared prohibitions. The other way round, conceptually starting with a totally open interface, the real problem is to minimally confine such a flow of information, until the wanted kind of confidentiality according to the declared prohibitions is achieved, thus still preserving a maximal availability.

We want to emphasize that we are not aiming at anything more. Neither do we want to confine the information owner in chatting about what he has in his human mind, nor do we want to hinder the communication partner in observing real-world facts, nor to prevent him from exploiting any further information source. We care only about the conceptual information flow from the internal state of a technical device to an interacting computing agent based on protocol-complying exchanges of messages. Clearly, the occurrences of such flows might depend on additional circumstances, as captured by our assumptions and postulates about the communication partner regarding his a priori knowledge and his background knowledge, respectively.

So, our achievements are as promising as these assumptions and postulates are realistic and, furthermore, all the other features left aside by us can also be suitably dealt with.

6.1.7 Selected Bibliographic Notes

The study of inference-proof interactions of a logic-oriented information system started with two contributions about the interaction of closed-query evaluation with response preparation. Sicherman, Jonge, and Riet [611] suggested the refusing approach early, and Bonatti, Kraus, and Subrahmanian [96] later introduced the lying approach. Following

a first attempt to compare the two approaches, Biskup and Bonatti [78] set up a unifying framework for further developments.

Biskup and Bonatti [75, 76] introduced the combination of refusing and lying, for closed queries. Furthermore, Biskup and Bonatti [77] treated open-query evaluation with response-set preparation for a decidable relational submodel; considerably much later, Biskup, Bring, and Bulinski [79] reported on a partial prototype implementation with some optimizations. Among other attempts to restrict the expressibility of relational a priori knowledge, queries and prohibitions to enable inference-proofness in the spirit of access control by means of SQL only, Biskup, Embley, and Lochner [71] identified the impact of relational database schema normalization. Biskup and Weibert [86] extended all three approaches to alterations of responses to closed queries to an underlying incomplete propositional information system, which offers a third option (*don't know*) in addition to *yes* and *no*. Biskup, Gogolin, Seiler, and Weibert [81] added knowledge update transaction as a further interaction under lying, later also treated for refusing. Moreover, Biskup and Tadros [84] investigated the impact of non-monotonic reasoning for the interaction of belief revision. Biskup and Wiese [87] studied a concept of view generation as yet another interaction under lying—essentially, a as a combination of a restricted first-order logic satisfiability problem and a minimization problem. Later Biskup, Dahn, Diekmann, Menzel, Schalge, and Wiese [80] presented a prototype implementation exploring several heuristic optimizations and approximations. Biskup and Preuß [83] invented another method for view generation, later also extended to view updating, essentially based on weakening by means of disjunctions of prohibitions. Biskup, Tadros, and Zarouali [85] explored how to handle interactions expressed as procedural program executions in an inference-proof way, in particular exploiting methods of language-based security aiming at the security policy of non-interference. Biskup and Preuß [82] analyzed the fragmentation approach to secure data outsourcing. Finally, Biskup [73] studied the interaction of closed-query evaluation and view generation in the framework of abstract data sources, already presenting the framework reused in Section 6.1.2.

Some of these developments are discussed by Biskup [74]. Our narrower topic is embedded in some more general streams of research and related to many other specific topics, as outlined in Biskup [72]. Guarnieri, Marinovic, and Basin [243] and Halpern and O'Neill [255] are examples of taking a wider perspective, that includes probabilistic considerations.

Bibliography

[1] 3GPP. *3GPP TS 29.520 - 5G System; Network Data Analytics Services;Stage 3*. Tech. rep. 29.520. Version 15.3.0. 3rd Generation Partnership Project (3GPP), Mar. 2019 (cit. on p. 332).

[2] 3GPP. *NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone*. Tech. rep. 3rd Generation Partnership Project (3GPP), TS 38.101-1 V16.6.0, 2021 (cit. on p. 376).

[3] 3GPP. *System Architecture for the 5G System*. Tech. rep. 3rd Generation Partnership Project (3GPP), TS 23.501, Release 15.11.0, Sept. 2020 (cit. on pp. 371, 372).

[4] 3GPP TSG GERAN. *TR 36.888 V12.0.0: Study on provision of low-cost Machine-Type Communications (MTC) USer Equipment(UEs) based on LTE*. Tech. rep. 3GPP, 2013 (cit. on p. 264).

[5] 3GPP TSG GERAN. *TR 45.880 V13.1.0: Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT) (Release 13)*. Tech. rep. 3GPP, 2015 (cit. on p. 263).

[6] 5GAA. *White paper: Making 5G proactive and predictive for the automotive industry*. Tech. rep. 5G Automotive Association, Jan. 2020 (cit. on p. 332).

[7] C. Abrams. "Google's effort to prevent blindness shows AI challenges". In: *The Wall Street Journal* 1 (2019), p. 26 (cit. on p. 17).

[8] N. Abramson. "THE ALOHA SYSTEM: another alternative for computer communications". In: *Procs. of the Joint Computer Conference 1970*. ACM Press, 1970, p. 281. URL: http://portal.acm.org/citation.cfm?doid=1478462.1478502 (cit. on p. 318).

[9] F. Adib and D. Katabi. "See through walls with WiFi!" In: *SIGCOMM Computer Communication Review* 43.4 (Aug. 2013), pp. 75–86 (cit. on p. 246).

[10] C. C. Aggarwal and S. Y. Philip. "A condensation approach to privacy preserving data mining". In: *Procs. of the Int. Conference on Extending Database Technology 2004*. Springer. 2004, pp. 183–199 (cit. on p. 15).

[11] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. "Fast Algorithms for Projected Clustering". In: *Procs. of the ACM SIGMOD Int. Conference on Management of Data 1999*. SIGMOD '99. New York, NY, USA: ACM, 1999, pp. 61–72 (cit. on p. 148).

[12] R. Agrawal and R. Srikant. "Privacy-preserving data mining". In: *Procs. of the ACM SIGMOD Int. Conference on Management of Data 2000*. 2000, pp. 439–450 (cit. on p. 15).

[13] D. Aha. "Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms". In: *International Journal of Man-Machine Studies* 36.2 (1992), pp. 267–287 (cit. on pp. 139, 152).

[14] M. A. Akhloufi, S. Arola, and A. Bonnet. "Drones chasing drones: Reinforcement learning and deep search area proposal". In: *Drones* 3.3 (2019), p. 58 (cit. on p. 222).

[15] L. D. et al. "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing". In: *Procs. IEEE INFOCOM 2012*. 2012 (cit. on p. 298).

[16] Y. A. A. S. Aldeen, M. Salleh, and M. A. Razzaque. "A comprehensive review on privacy preserving data mining". In: *SpringerPlus* 4.1 (2015), pp. 1–36 (cit. on p. 15).

[17] S. Ali et al. *6G White Paper on Machine Learning in Wireless Communication Networks*. Apr. 2020. URL: https://arxiv.org/abs/2004.13875 (cit. on pp. 274, 332).

[18] S. Aljawarneh, A. Anguera, J. W. Atwood, J. A. Lara, and D. Lizcano. "Particularities of data mining in medicine: lessons learned from patient medical time series data analysis". In: *EURASIP Journal on Wireless Communications and Networking* 2019.1 (2019), pp. 1–29 (cit. on p. 15).

[19] M. Allhoff, A. Schönhuth, M. Martin, I. G. Costa, S. Rahmann, and T. Marschall. "Discovering motifs that induce sequencing errors". In: *BMC Bioinformatics* 14 Suppl 5 (2013), S1 (cit. on p. 55). **SFB876-B1**

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

https://doi.org/10.1515/9783110785982-007

[20] Y. Altintas, P. Kersting, D. Biermann, E. Budak, B. Denkena, and I. Lazoglu. “Virtual process systems for part machining operations”. In: *CIRP Annals* 63.2 (2014), pp. 585–605 (cit. on p. 157).

[21] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi. “UAV assistance paradigm: State-of-the-art in applications and challenges”. In: *Journal of Network and Computer Applications* 166 (2020), p. 102706 (cit. on p. 213).

[22] S. Amal, F. Felix, B. Jens, M. Katharina, W. Petra, and R. Wolfgang. “Simulation and Sensor Data Fusion for Machine Learning Application”. In: *Advanced Engineering Informatics* (2022) (cit. on p. 166).

[23] X. An, Q. Yang, and P. M. Bentler. “A latent factor linear mixed model for high-dimensional longitudinal data analysis”. In: *Statistics in medicine* 32.24 (2013), pp. 4229–4239 (cit. on p. 9).

[24] M. W. Anderson and I. Schrijver. “Next Generation DNA Sequencing and the Future of Genomic Medicine”. In: *Genes* 1.1 (June 2010). Number: 1Publisher: Molecular Diversity Preservation Int., pp. 38–69. URL: https://www.mdpi.com/2073-4425/1/1/38 (cit. on p. 55).

[25] D. W. Apley and J. Zhu. “Visualizing the effects of predictor variables in black box supervised learning models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82.4 (2020), pp. 1059–1086 (cit. on p. 17).

[26] S. Arabi, E. Sabir, H. Elbiaze, and M. Sadik. “Data gathering and energy transfer dilemma in UAV-assisted flying access network for IoT”. In: *Sensors* 18.5 (2018), p. 1519 (cit. on p. 213).

[27] L. Archibald, L. Phillips, D. Monnet, J. E. McGowan Jr, F. Tenover, and R. Gaynes. “Antimicrobial resistance in isolates from inpatients and outpatients in the United States: increasing importance of the intensive care unit”. In: *Clinical Infectious Diseases* 24.2 (1997), pp. 211–215 (cit. on p. 7).

[28] D. Ardila et al. “End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography”. In: *Nature medicine* 25.6 (2019), pp. 954–961 (cit. on pp. 5, 8).

[29] C. Arendt. “Raw environmental indoor sensor data”. In: (Nov. 2019). URL: http://doi.org/10.5281/zenodo.3555406 (cit. on p. 324).

[30] C. Arendt, S. Böcker, and C. Wietfeld. “Data-Driven Model-Predictive Communication for Resource-Efficient IoT Networks”. In: *Procs. of the IEEE World Forum on Internet of Things 2020*. Apr. 2020 (cit. on pp. 270, 323–328). **SFB876-A4**

[31] V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. “On the Power of Color Refinement”. In: *Procs. of the Int. Symposium on Fundamentals of Computation Theory 2015*. 2015, pp. 339–350 (cit. on p. 91).

[32] A. Asuncion and D. J. Newman. *UCI Machine Learning Repository*. 2007 (cit. on pp. 140, 152).

[33] J.-Y. Audibert and R. Munos. “Introduction to bandits: Algorithms and theory”. In: *ICML Tutorial on bandits* (2011) (cit. on p. 236).

[34] L. Babai and L. Kucera. “Canonical labelling of graphs in linear average time”. In: *Procs. of the Symposium on Foundations of Computer Science 1979*. IEEE, 1979, pp. 39–46 (cit. on p. 91).

[35] D. Baek, M. Hwang, H. Kim, and D.-S. Kwon. “Path planning for automation of surgery robot based on probabilistic roadmap and reinforcement learning”. In: *Procs. of the Int. Conference on Ubiquitous Robots 2018*. IEEE. 2018, pp. 342–347 (cit. on p. 6).

[36] B. M. Bai, B. Nalini, and J. Majumdar. “Analysis and detection of diabetes using data mining techniques – a big data application in health care”. In: *Emerging Research in Computing, Information, Communication and Applications*. Springer, 2019, pp. 443–455 (cit. on p. 15).

[37] L. Bai, L. Rossi, Z. Zhang, and E. R. Hancock. “An Aligned Subtree Kernel for Weighted Graphs”. In: *Procs. of the Int. Conference on Machine Learning 2015*. 2015, pp. 30–39 (cit. on p. 89).

[38] S. Bai, J. Z. Kolter, and V. Koltun. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In: *arXiv: Computing Research Repository* (2018). DOI: arXiv:1803.01271 (cit. on p. 11).

[39] C. A. Balanis. *Antenna Theory: Analysis and Design*. 4th ed. Wiley, Feb. 2016 (cit. on p. 376).

[40] W. Balid, H. Tafish, and H. H. Refai. “Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.6 (2018), pp. 1784–1794 (cit. on p. 245).

[41] A. Banks et al. “MQTT Version 3.1.1”. In: *OASIS standard* (2014) (cit. on p. 265).

[42] P. Bansal and K. M. Kockelman. “Forecasting Americans’ long-term adoption of connected and autonomous vehicle technologies”. In: *Transportation Research Part A: Policy and Practice* 95.8 (2017), pp. 49–63 (cit. on p. 293).

[43] M. K. Baowaly, C.-C. Lin, C.-L. Liu, and K.-T. Chen. “Synthesizing electronic health records using improved generative adversarial networks”. In: *Journal of the American Medical Informatics Association* 26.3 (2019), pp. 228–241 (cit. on p. 16).

[44] A. S. Barkousaraie, O. Ogunmolu, S. Jiang, and D. Nguyen. “A Fast Deep Learning Approach for Beam Orientation Optimization for Prostate Cancer IMRT Treatments”. In: *arXiv: Physics* (2019). DOI: arXiv:1905.00523 (cit. on p. 8).

[45] J. A. Barta, C. A. Powell, and J. P. Wisnivesky. “Global epidemiology of lung cancer”. In: *Annals of global health* 85.1 (2019) (cit. on p. 7).

[46] E. Basar, M. Di Renzo, J. De Rosny, M. Debbah, M.-S. Alouini, and R. Zhang. “Wireless Communications Through Reconfigurable Intelligent Surfaces”. In: *IEEE Access* 7 (2019), pp. 116753–116773 (cit. on p. 385).

[47] H. Bast, J. Sternisko, and S. Storandt. “Delay-robustness of transfer patterns in public transportation route planning”. In: *Procs. of the Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems 2013*. Vol. 33. Schloss Dagstuhl Leibniz-Zentrum fuer Informatik. 2013, pp. 42–54 (cit. on pp. 234, 235).

[48] H. Bast et al. “Fast routing in very large public transportation networks using transfer patterns”. In: *Procs. of the European Symposium on Algorithms 2010*. Springer. 2010, pp. 290–301 (cit. on pp. 234, 235).

[49] H. Bast et al. *Route Planning in Transportation Networks*. Tech. rep. MSR-TR-2014-4. Jan. 2014 (cit. on p. 233).

[50] G. E. Batista and M. C. Monard. “An analysis of four missing data treatment methods for supervised learning”. In: *Applied Artificial Intelligence* 17.5-6 (2003), pp. 519–533 (cit. on p. 13).

[51] S. Batra, Z. Huang, A. Petrenko, T. Kumar, A. Molchanov, and G. S. Sukhatme. “Decentralized control of quadrotor swarms with end-to-end deep reinforcement learning”. In: *Procs. of the Conference on Robot Learning 2022*. PMLR. 2022, pp. 576–586 (cit. on p. 222).

[52] É. Beke, A. Bódi, T. G. Katalin, T. Kovács, D. Maros, and L. Gáspár. “The role of drones in linking industry 4.0 and ITS Ecosystems”. In: *Procs. of the IEEE Int. Symposium on Computational Intelligence and Informatics 2018*. IEEE. 2018, pp. 000191–000198 (cit. on p. 212).

[53] C. Bektas, S. Böcker, F. Kurtz, and C. Wietfeld. “Reliable Software-Defined RAN Network Slicing for Mission-Critical 5G Communication Networks”. In: *Procs. of the IEEE Globecom Workshops 2019*. Waikoloa, Hawaii, USA, Dec. 2019. URL: https://ieeexplore.ieee.org/document/9024677 (cit. on p. 366).

[54] C. Bektas, D. Overbeck, and C. Wietfeld. “SAMUS: Slice-Aware Machine Learning-based Ultra-Reliable Scheduling”. In: *Procs. of the IEEE Int. Conference on Communications 2021*. Montreal, Canada, June 2021 (cit. on pp. 364–367, 369–371, 373). **SFB876-B4**

[55] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. “Support Vector Clustering”. In: *Journal of Machine Learning Research* 2 (Mar. 2002), pp. 125–137 (cit. on p. 148).

[56] J. L. Bentley and J. B. Saxe. “Decomposable Searching Problems I: Static-to-Dynamic Transformation”. In: *The Journal of Algorithms* 1.4 (1980), pp. 301–358 (cit. on p. 66).

[57] J. Bergstra and Y. Bengio. “Random search for hyper-parameter optimization.” In: *Journal of Machine Learning Research* 13.2 (2012) (cit. on p. 176).

[58] M. Bernas, B. Płaczek, and W. Korski. “Wireless network with Bluetooth low energy beacons for vehicle detection and classification”. In: *Procs. of the Int. Conference on Computer Networks 2018*. Ed. by P. Gaj, M. Sawicki, G. Suchacka, and A. Kwiecień. Cham: Springer Int. Publishing, 2018, pp. 429–444 (cit. on p. 246).

[59] H. M. Bette, L. Habel, T. Emig, and M. Schreckenberg. “Mechanisms of jamming in the Nagel-Schreckenberg model for traffic flow”. In: *Physical Review E* 95.012311 (Jan. 2017). URL: http://dx.doi.org/10.1103/PhysRevE.95.012311 (cit. on p. 287). **SFB876-B4**

[60] A. Bharti, R. Adeogun, and T. Pedersen. “Learning Parameters of Stochastic Radio Channel Models From Summaries”. In: *IEEE Open Journal of Antennas and Propagation* 1 (2020), pp. 175–188 (cit. on p. 274).

[61] S. S. Bhowmick and B. Seah. “Clustering and Summarizing Protein-Protein Interaction Networks: A Survey”. In: *IEEE Transactions on Knowledge Data Engineering* 28.3 (2016), pp. 638–658 (cit. on p. 86).

[62] G. Bhuvaneswari and R. Annamalai. “Development of a Solar Cell Model in MATLAB for PV Based Generation System”. In: *Procs. of the IEEE India Conference 2011*. 2011, pp. 1–5 (cit. on p. 197).

[63] S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. “Multi-Task Learning for HIV Therapy Screening”. In: *Procs. of the Int. Conference on Machine Learning 2008*. ICML ’08. New York, NY, USA: ACM, 2008, pp. 56–63 (cit. on p. 78).

[64] F. Biessmann, D. Salinas, S. Schelter, P. Schmidt, and D. Lange. “’Deep’ Learning for Missing Value Imputation in Tables with Non-Numerical Data”. In: *Procs. of the ACM Int. Conference on Information and Knowledge Management 2018*. 2018, pp. 2017–2025 (cit. on p. 13).

[65] M. Binder, F. Pfisterer, M. Lang, L. Schneider, L. Kotthoff, and B. Bischl. “mlr3pipelines - Flexible Machine Learning Pipelines in R”. In: *Journal of Machine Learning Research* 22.184 (2021), pp. 1–7. URL: http://jmlr.org/papers/v22/21-0281.html (cit. on p. 83).

[66] M. Birgersson and G. Hansson. “Data integration using machine learning: Automation of data mapping using machine learning techniques”. MA thesis. 2016 (cit. on p. 12).

[67] B. Bischl, M. Lang, O. Mersmann, J. Rahnenführer, and C. Weihs. “BatchJobs and BatchExperiments: Abstraction Mechanisms for Using R in Batch Environments”. In: *Journal of Statistical Software* 64.11 (2015), pp. 1–235. URL: http://www.jstatsoft.org/v64/i11 (cit. on p. 83). **SFB876-A3**

[68] B. Bischl et al. “Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges”. In: *arXiv: Statistics* (2021). DOI: arXiv:2107.05847v2 (cit. on p. 74). **SFB876-A3**

[69] B. Bischl et al. “mlr: Machine Learning in R”. In: *Journal of Machine Learning Research* 17.170 (2016), pp. 1–5. URL: http://jmlr.org/papers/v17/15-066.html (cit. on p. 82). **SFB876-A3**

[70] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006 (cit. on p. 221).

[71] J. Biskup, D. Embley, and J.-H. Lochner. “Reducing inference control to access control for normalized database schemas”. In: *Information Processing Letters* 106 (2008), pp. 8–12 (cit. on p. 412).

[72] J. Biskup. “Inference Control”. In: *Encyclopedia of Cryptography, Security and Privacy*. Ed. by S. Jajodia, P. Samarati, and M. Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 1–7. URL: https://doi.org/10.1007/978-3-642-27739-9_753-2 (cit. on p. 412).

[73] J. Biskup. “Inference-Proof Monotonic Query Evaluation and View Generation Reconsidered”. In: *Procs. of Data and Applications Security and Privacy 2020*. Ed. by A. Singhal and J. Vaidya. Vol. 12122. LNCS. Springer, 2020, pp. 79–99 (cit. on pp. 394, 412).

[74] J. Biskup. “Selected Results and Related Issues of Confidentiality-Preserving Controlled Interaction Execution”. In: *Procs. of Foundations of Information and Knowledge Systems 2016*. Ed. by M. Gyssens and G. R. Simari. Vol. 9616. LNCS. Springer, 2016, pp. 211–234 (cit. on p. 412). **SFB876-A5**

[75] J. Biskup and P. A. Bonatti. “Controlled query evaluation for enforcing confidentiality in complete information systems”. In: *The International Journal of Information Security* 3.1 (2004), pp. 14–27 (cit. on p. 412).

[76] J. Biskup and P. A. Bonatti. “Controlled Query Evaluation for Known Policies by Combining Lying and Refusal”. In: *Annals of Mathematics and Artificial Intelligence* 40.1-2 (2004), pp. 37–62 (cit. on p. 412).

[77] J. Biskup and P. A. Bonatti. “Controlled query evaluation with open queries for a decidable relational submodel”. In: *Annals of Mathematics and Artificial Intelligence* 50.1-2 (2007), pp. 39–77 (cit. on p. 412).

[78] J. Biskup and P. A. Bonatti. “Lying versus refusal for known potential secrets”. In: *Data & Knowledge Engineering* 38.2 (2001), pp. 199–222 (cit. on p. 412).

[79] J. Biskup, M. Bring, and M. Bulinski. “Inference control of open relational queries under closed-world semantics based on theorem proving”. In: *The Information Systems Journal* 70 (2017), pp. 32–47 (cit. on p. 412).

[80] J. Biskup, C. Dahn, K. Diekmann, R. Menzel, D. Schalge, and L. Wiese. “Publishing inference-proof relational data: An implementation and experiments”. In: *Data & Knowledge Engineering* 120 (2019), pp. 1–44 (cit. on p. 412).

[81] J. Biskup, C. Gogolin, J. Seiler, and T. Weibert. “Inference-proof view update transactions with forwarded refreshments”. In: *Journal of Computer Security* 19 (2011), pp. 487–529 (cit. on p. 412).

[82] J. Biskup and M. Preuß. “Checking Inference-Proofness of Attribute-Disjoint and Duplicate-Preserving Fragmentations”. In: *Annals of Mathematics and Artificial Intelligence* (2019) (cit. on p. 412).

[83] J. Biskup and M. Preuß. “Information Control by Policy-Based Relational Weakening Templates”. In: *Procs. of the European Symposium on Research in Computer Security 2016*. Ed. by I. G. Askoxylakis, S. Ioannidis, S. K. Katsikas, and C. A. Meadows. Vol. 9879. LNCS. Springer, 2016, pp. 361–381 (cit. on p. 412).

[84] J. Biskup and C. Tadros. “Preserving confidentiality while reacting on iterated queries and belief revisions”. In: *Annals of Mathematics and Artificial Intelligence* 73.1-2 (2015), pp. 75–123 (cit. on p. 412). **SFB876-A5**

[85] J. Biskup, C. Tadros, and J. Zarouali. “Confidentiality enforcement by hybrid control of information flows”. In: *arXiv: Computing Research Repository* (2017). DOI: arXiv:1707.08482 (cit. on p. 412).

[86] J. Biskup and T. Weibert. “Keeping secrets in incomplete databases”. In: *International Journal of Information Security* 7.3 (2008), pp. 199–217 (cit. on p. 412).

[87] J. Biskup and L. Wiese. “A sound and complete model-generation procedure for consistent and confidentiality-preserving databases”. In: *Theoretical Computer Science* 412.31 (2011), pp. 4044–4072 (cit. on p. 412).

[88] A. Blagojević et al. “Artificial intelligence approach towards assessment of condition of COVID-19 patients-Identification of predictive biomarkers associated with severity of clinical condition and disease progression”. In: *Computers in Biology and Medicine* 138 (2021), p. 104869 (cit. on p. 11).

[89] A. Bloch, D. Berger, and J. Takala. “Understanding circulatory failure in sepsis”. In: *Intensive Care Medicine* 42.12 (2016), pp. 2077–2079 (cit. on p. 5).

[90] D. B. Blumenthal, N. Boria, J. Gamper, S. Bougleux, and L. Brun. “Comparing heuristics for graph edit distance computation”. In: *VLDB – The International Journal on Very Large Data Bases* 29.1 (2020), pp. 419–458. URL: https://doi.org/10.1007/s00778-019-00544-1 (cit. on pp. 88, 94).

[91] C. Bock, T. Gumbsch, M. Moor, B. Rieck, D. Roqueiro, and K. Borgwardt. “Association mapping in biomedical time series via statistically significant shapelet mining”. In: *Bioinformatics* 34.13 (2018), pp. i438–i446 (cit. on pp. 9, 15).

[92] C. Bock, M. Moor, C. R. Jutzeler, and K. Borgwardt. “Machine learning for biomedical time series classification: from shapelets to deep learning”. In: *Artificial Neural Networks*. Springer, 2021, pp. 33–71 (cit. on p. 9).

[93] S. Böcker, C. Arendt, P. Jörke, and C. Wietfeld. “LPWAN in the Context of 5G: Capability of LoRaWAN to Contribute to mMTC”. In: *Procs. of the IEEE World Forum on Internet of Things 2019*. IEEE, 2019 (cit. on pp. 262, 316, 317, 319, 320).

[94] A. Bommert and M. Lang. “stabm: Stability Measures for Feature Selection”. In: *Journal of Open Source Software* 6.59 (2021), p. 3010. URL: https://doi.org/10.21105/joss.03010 (cit. on p. 83). **SFB876-A3**

[95] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang. “Benchmark for Filter Methods for Feature Selection in High-Dimensional Classification Data”. In: *Computational Statistics & Data Analysis* 143 (Mar. 2020), p. 106839 (cit. on pp. 72, 81, 82). **SFB876-A3**

[96] P. A. Bonatti, S. Kraus, and V. S. Subrahmanian. “Foundations of Secure Deductive Databases”. In: *IEEE Transactions on Knowledge and Data Engineering* 7.3 (1995), pp. 406–422 (cit. on p. 411).

[97] M. Borowski, M. Görges, R. Fried, O. Such, C. Wrede, and M. Imhoff. “Medical device alarms”. In: (2011) (cit. on p. 5).

[98] G. E. P. Box, G. Jenkins, and G. C. Reinsel. *Time Series Analysis. Forecasting and Control*. 3rd. Englewood Cliffs: Prentice Hall, 1994 (cit. on p. 119).

[99] J. Boyan and M. Littman. “Packet routing in dynamically changing networks: A reinforcement learning approach”. In: *Advances in Neural Information Processing Systems 6: Procs. of the 1993 Conference*. 1994, pp. 671–678 (cit. on pp. 274, 275).

[100] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000) (cit. on p. 30).

[101] L. Breiman. “Bagging Predictors”. In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140. URL: https://doi.org/10.1023/A:1018054314350 (cit. on pp. 125, 169, 171).

[102] L. Breiman. “Random Forests”. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32 (cit. on pp. 17, 152, 245).

[103] H. Breiteneder et al. “Biomarkers for diagnosis and prediction of therapy responses in allergic diseases and asthma”. In: *Allergy* 75.12 (2020), pp. 3039–3068 (cit. on p. 9).

[104] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. “LOF: Identifying Density-Based Local Outliers”. In: *Procs. of the ACM SIGMOD Int. Conference on Management of Data 2000*.

SIGMOD '00. New York, NY, USA: Association for Computing Machinery, 2000, pp. 93–104. URL: https://doi.org/10.1145/342009.335388 (cit. on p. 14).

[105] A. Z. Broder. "On the resemblance and containment of documents". In: *Procs. of the Compression and Complexity of SEQUENCES 1997*. June 1997, pp. 21–29 (cit. on pp. 89, 93).

[106] G. Brown, J. L. Wyatt, and P. Tino. "Managing Diversity in Regression Ensembles". In: *Journal of Machine Learning Research* (2005), pp. 1621–1650 (cit. on p. 169).

[107] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer. "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques". In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1790–1821 (cit. on pp. 275, 332).

[108] N. Bui and J. Widmer. "OWL: A Reliable Online Watcher for LTE Control Channel Measurements". In: *Procs. of the Workshop on All Things Cellular: Operations, Applications and Challenges 2016*. ATC '16. New York, NY, USA: ACM, 2016, pp. 25–30 (cit. on pp. 345, 347).

[109] H. Bunke. "On a relation between graph edit distance and maximum common subgraph". In: *Pattern Recognition Letters* 18.8 (1997), pp. 689–694 (cit. on p. 87).

[110] J. Burridge. "Empirical Bayes analysis of survival time data". In: *Journal of the Royal Statistical Society. Series B* 43.1 (1981), pp. 65–75 (cit. on p. 63).

[111] S. Buschjäger, T. Liebig, and K. Morik. "Gaussian Model Trees for Traffic Imputation". In: *Procs. of the Int. Conference on Pattern Recognition Applications and Methods 2019*. SciTePress. 2019, pp. 243–254. URL: https://www.scitepress.org/PublicationsDetail.aspx?ID=g+tVIY+KNts=%5C&t=1 (cit. on p. 229). **SFB876-A1, SFB876-B4**

[112] M. D. Butti, H. Chanfreau, D. Martinez, D. García, E. Lacunza, and M. C. Abba. "BioPlat: a software for human cancer biomarker discovery". In: *Bioinformatics* 30.12 (2014), pp. 1782–1784 (cit. on p. 9).

[113] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. "A Limited Memory Algorithm for Bound Constrained Optimization". In: *SIAM Journal on Scientific Computing (SISC)* 16.5 (Sept. 1995), pp. 1190–1208 (cit. on p. 168).

[114] D. L. Cameron et al. "GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly". In: *Genome Research* 27.12 (Dec. 2017), pp. 2050–2060. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5741059/ (cit. on p. 57).

[115] K. S. Candan, R. Rossini, M. L. Sapino, and X. Wang. "sDTW: computing DTW distances using locally relevant constraints based on salient feature alignments". In: *arXiv: Computing Research Repository* (2012). DOI: arXiv:1208.0275 (cit. on p. 122).

[116] B.-T. Cao, S. Freitag, and G. Meschke. "A Hybrid RNN-GPOD Surrogate Model for Real-time Settlement Predictions in Mechanised Tunnelling". In: *Advanced Modeling and Simulation in Engineering Sciences* 3.1 (Mar. 2016), p. 5. URL: https://link.springer.com/article/10.1186/s40323-016-0057-9 (cit. on pp. 160, 161).

[117] H. Carr, J. Snoeyink, and U. Axen. "Computing contour trees in all dimensions". In: *Computational Geometry* 24.2 (2003), pp. 75–94 (cit. on p. 16).

[118] A. Cassini et al. "Attributable deaths and disability-adjusted life-years caused by infections with antibiotic-resistant bacteria in the EU and the European Economic Area in 2015: a population-level modelling analysis". In: *The Lancet Infectious Diseases* 19.1 (2019), pp. 56–66 (cit. on p. 7).

[119] E. R. Cavalcanti, J. A. R. de Souza, M. A. Spohn, R. C. d. M. Gomes, and A. F. B. F. d. Costa. "VANETs' research over the past decade: Overview, credibility, and trends". In: *SIGCOMM Computer Communication Review* 48.2 (May 2018), pp. 31–39 (cit. on pp. 274, 280).

[120] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006 (cit. on pp. 137, 144).

[121] P. Chapman et al. *CRISP–DM 1.0*. Tech. rep. The CRISP–DM Consortium, Aug. 2000 (cit. on p. 115).

[122] S. Chatterjee and A. S. Hadi. "Sensitivity analysis in linear regression". In: (1988) (cit. on p. 60).

[123] A. Chaudhary and S. S. Singh. "Lung cancer detection on CT images by using image processing". In: *Procs. of the Int. Conference on Computing Sciences 2012*. IEEE. 2012, pp. 142–146 (cit. on p. 5).

[124] H. Chaudhary. "Application of the Theory of a Single First Order Equation to Traffic Flow". In: *Journal of the Institute of Engineering* 9.1 (2014), pp. 175–181 (cit. on p. 237).

[125] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-Sampling Technique". In: *Journal of Artificial Intelligence Research* 16.1 (June 2002), pp. 321–357 (cit. on p. 125).

[126] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer. "SMOTEBoost: Improving Prediction of the Minority Class in Boosting". In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2003*. 2003 (cit. on p. 125).

[127] M. K. Cheezum, W. F. Walker, and W. H. Guilford. "Quantitative Comparison of Algorithms for Tracking Single Fluorescent Particles". In: *Biophysical Journal* 81.4 (2001), pp. 2378–2388. URL: https://www.sciencedirect.com/science/article/pii/S0006349501758845 (cit. on p. 28).

[128] J. Chen and B. Huang. "An In-Process Neural Network-Based Surface Roughness Prediction (INN-SRP) System Using a Dynamometer in End Milling Operations". In: *The Int. Journal of Advanced Manufacturing Technology* 21.5 (2003), pp. 339–347 (cit. on pp. 158, 160).

[129] J. Chen, S. Li, B. Dai, and G. Zhou. "Active Learning for Age Regression in Social Media". In: *Procs. of the Int. Symposium on Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data 2016*. Ed. by M. Sun, X. Huang, H. Lin, Z. Liu, and Y. Liu. Cham: Springer Int. Publishing, 2016, pp. 351–362 (cit. on pp. 162, 163).

[130] K. Chen and L. Liu. "Privacy preserving data classification with rotation perturbation". In: *Procs. of the IEEE Int. Conference on Data Mining 2005*. IEEE. 2005, 4–pp (cit. on p. 15).

[131] S. Chen, B. Liu, M. Qian, and C. Zhang. "Kernel k-Means Based Framework For Aggregate Outputs Classification". In: *Procs. of the Int. Conference on Data Mining Workshops 2009*. 2009, pp. 356–361 (cit. on pp. 141, 151).

[132] X. Chen, H. Huo, J. Huan, and J. S. Vitter. "An efficient algorithm for graph edit distance computation". In: *Knowledge-Based Systems* 163 (2019), pp. 762–775 (cit. on pp. 88, 93, 94).

[133] Z. Chen, W. Chen, and Y. Shi. "Ensemble learning with label proportions for bankruptcy prediction". In: *Expert Systems with Applications* 146 (2020), pp. 113–155 (cit. on p. 139).

[134] C.-C. Chiu and C.-Y. Tsai. "A k-anonymity clustering method for effective data privacy preservation". In: *Procs. of the Int. Conference on Advanced Data Mining and Applications 2007*. Springer. 2007, pp. 89–99 (cit. on p. 15).

[135] K. Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *Procs. of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: arXiv:1406.1078 (cit. on p. 11).

[136] R. Chrobok, O. Kaumann, J. Wahle, and M. Schreckenberg. "Three Categories of Traffic Data: Historical, Current, and Predictive". In: *IFAC Symposium on Control in Transportation Systems 2000*. Vol. 33. 9. 2000, pp. 221–226. URL: https://www.sciencedirect.com/science/article/pii/S1474667017381508 (cit. on p. 289).

[137] H. M. Chung and P. Gray. "Data mining". In: *Journal of Management Information Systems* 16.1 (1999), pp. 11–16 (cit. on p. 15).

[138] Y. Chung, S. Krishnan, and T. Kraska. “A data quality metric (DQM): how to estimate the number of undetected errors in data sets”. In: *Procs. of the VLDB Endowment* 10.10 (2016), pp. 1094–1105 (cit. on p. 12).

[139] I. Cima et al. “Cancer genetics-guided discovery of serum biomarker signatures for diagnosis and prognosis of prostate cancer”. In: *Procs. of the National Academy of Sciences* 108.8 (2011), pp. 3342–3347 (cit. on pp. 9, 10).

[140] R. Clark Robertson and Tri T Ha. “Effect of capture on throughput of variable length packet Aloha systems”. In: *Computer Communications* 17.12 (1994), pp. 836–842 (cit. on p. 318).

[141] W. S. Cleveland and S. J. Devlin. “Locally Weighted Regression : An Approach to Regression Analysis by Local Fitting Locally Weighted Regression : An Approach to Regression Analysis by Local Fiiting”. In: *Journal of the American Statistical Association* 83.403 (1988), pp. 37–41 (cit. on p. 324).

[142] H. Climente-González, C.-A. Azencott, S. Kaski, and M. Yamada. “Block HSIC Lasso: model-free biomarker detection for ultra-high dimensional data”. In: *Bioinformatics* 35.14 (2019), pp. i427–i435 (cit. on p. 9).

[143] L. Codeca, R. Frank, and T. Engel. “Luxembourg SUMO traffic (LuST) scenario: 24 hours of mobility for vehicular networking research”. In: *Procs. of the Vehicular Networking Conference 2015*. IEEE. 2015, pp. 1–8 (cit. on p. 239).

[144] D. Cohen-Steiner, H. Edelsbrunner, and D. Morozov. “Vines and vineyards by updating persistence in linear time”. In: *Procs. of the Symposium on Computational Geometry 2006*. 2006, pp. 119–126 (cit. on p. 16).

[145] B. Coifman and S. Kim. “Speed estimation and length based vehicle classification from freeway single-loop detectors”. In: *Transportation Research Part C: Emerging Technologies* 17.4 (2009), pp. 349–364. URL: https://www.sciencedirect.com/science/article/pii/S0968090X09000072 (cit. on p. 245).

[146] I. Corporation. *SLMD960H12L: IXOLAR High Efficiency SolarMD*. 2010 (cit. on p. 201).

[147] C. Cortes and V. N. Vapnik. “Support–Vector Networks”. In: *Machine Learning Journal* 20 (Sept. 1995), pp. 273–297 (cit. on p. 245).

[148] C. Cox. *An Introduction to 5G*. Wiley, Feb. 2021 (cit. on p. 377).

[149] D. R. Cox. “Regression Models and Life-Tables”. In: *Journal of the Royal Statistical Society. Series B* 34.2 (1972), pp. 187–220 (cit. on pp. 62, 63, 72, 79).

[150] R. Crane. “Prediction of Attenuation by Rain”. In: *IEEE Transactions on Communications* 28.9 (1980), pp. 1717–1733 (cit. on p. 246).

[151] D. Cristiani, F. Bottonelli, A. Trotta, and M. Di Felice. “Inventory Management through Mini-Drones: Architecture and Proof-of-Concept Implementation”. In: *Procs. of the IEEE Int. Symposium on A World of Wireless, Mobile and Multimedia Networks*. 2020, pp. 317–322 (cit. on p. 213).

[152] K. Crombecq, L. De Tommasi, D. Gorissen, and T. Dhaene. “A novel sequential design strategy for global surrogate modeling”. In: *Procs. of the Winter Simulation Conference 2009*. IEEE. 2009, pp. 731–742 (cit. on p. 17).

[153] B. Crowther. “Flocking of autonomous unmanned air vehicles”. In: *The Aeronautical Journal* 107.1068 (2003), pp. 99–109 (cit. on pp. 214, 215).

[154] L. E. Cue la Rosa and D. A. Borges Oliveira. “Learning from Label Proportions with Prototypical Contrastive Clustering”. In: *Procs. of the AAAI Conference on Artificial Intelligence 2022*. AAAI. 2022 (cit. on p. 139).

[155] F. Cus, M. Milfelner, and J. Balic. “An intelligent system for monitoring and optimization of ball-end milling process”. In: *Journal of Materials Processing Technology* 175.1-3 (2006), pp. 90–97 (cit. on p. 115).

[156] M. Dalla Mura, S. Prasad, F. Pacifici, P. Gamba, J. Chanussot, and J. A. Benediktsson. “Challenges and opportunities of multimodality and data fusion in remote sensing”. In: *Procs. of the IEEE* 103.9 (2015), pp. 1585–1601 (cit. on p. 115).

[157] C. Daniel and L. Mary. “Fusion of audio visual cues for vehicle classification”. In: *Procs. of the Int. Conference on Next Generation Intelligent Systems 2016*. 2016, pp. 1–4 (cit. on p. 245).

[158] C. A. L. Dantec et al. “A tale of two publics: Democratizing design at the margins”. In: *Procs. of the Conference on Designing Interactive Systems 2010*. 2010 (cit. on p. 298).

[159] R. Dara, S. Kremer, and D. Stacey. “Clustering unlabeled data with SOMs improves classification of labeled real-world data”. In: *Procs. of the Int. Joint Conference on Neural Networks 2002*. Vol. 3. 2002, pp. 2237–2242 (cit. on p. 139).

[160] G. Das, D. Gunopulos, and H. Mannila. “Finding Similar Time Series”. In: *Procs. of the Symposium on Principles of Data Mining and Knowledge Discovery 1997*. 1997, pp. 88–100 (cit. on p. 121).

[161] T. Davenport and R. Kalakota. “The potential for artificial intelligence in healthcare”. In: *Future Healthcare Journal* 6.2 (2019), p. 94 (cit. on p. 4).

[162] S. De, Y. Hu, V. V. Meduri, Y. Chen, and S. Kambhampati. “Bayeswipe: A scalable probabilistic framework for improving data quality”. In: *Journal of Data and Information Quality* 8.1 (2016), pp. 1–30 (cit. on p. 12).

[163] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197 (cit. on p. 35).

[164] W. Delfmann, M. ten Hompel, W. Kersten, T. Schmidt, and W. Stölzle. “Logistics as a Science – Central Research Questions in the Era of the Fourth Industrial Revolution”. In: 11.1 (). URL: https://doi.org/10.23773/2018_9 (cit. on p. 110).

[165] D. Delling, T. Pajor, and R. Werneck. “Round-Based Public Transit Routing”. In: *Procs. of the Meeting on Algorithm Engineering and Experiments 2012*. Society for Industrial and Applied Mathematics, Jan. 2012 (cit. on p. 235).

[166] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosmanoglu. “Crowdsourced sensing and collaboration using twitter”. In: *Procs. of the IEEE Int. Symposium on A World of Wireless, Mobile and Multimedia Networks 2010*. 2010 (cit. on p. 298).

[167] A. Demiriz, K. Bennett, K. P. Bennett, and M. J. Embrechts. “Semi-supervised clustering using genetic algorithms”. In: *Procs. of Artificial Neural Networks in Engineering 1999*. ASME Press, 1999, pp. 809–814 (cit. on p. 139).

[168] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the Royal Statistical Society. Series B* (1977), pp. 1–38 (cit. on pp. 141, 148).

[169] J. Demšar. “Statistical Comparisons of Classifiers over Multiple Data Sets”. In: *The Journal of Machine Learning Research* 7 (2006), pp. 1–30 (cit. on p. 153).

[170] B. Denkena, M.-A. Dittrich, and F. Uhlich. “Augmenting Milling Process Data for Shape Error Prediction”. In: *Procs. of the CIRP Conference on Manufacturing Systems 2016*. Vol. 57. 2016, pp. 487–491 (cit. on p. 158).

[171] M. A. DePristo et al. “A framework for variation discovery and genotyping using next-generation DNA sequencing data”. In: *Nature Genetics* 43.5 (May 2011). Number: 5Publisher: Nature Publishing Group, pp. 491–498. URL: https://www.nature.com/articles/ng.806 (cit. on pp. 48, 51, 55).

[172] A. Derrow-Pinion et al. “ETA Prediction with Graph Neural Networks in Google Maps”. In: *Procs. of the ACM Int. Conference on Information and Knowledge Management 2021*. 2021 (cit. on p. 240).

[173] G. Deshpande, L. Libero, K. R. Sreenivasan, H. Deshpande, and R. K. Kana. “Identification of neural connectivity signatures of autism using machine learning”. In: *Frontiers in Human Neuroscience* 7 (2013), p. 670 (cit. on p. 10).

[174] I. S. Dhillon, Y. Guan, and B. Kulis. “Kernel k-means: spectral clustering and normalized cuts”. In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2004*. Ed. by W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel. SIGKDD ’04. New York, USA: ACM Press, 2004, pp. 551–556 (cit. on pp. 141, 148).

[175] J. Dibbelt, T. Pajor, B. Strasser, and D. Wagner. “Intriguingly simple and fast transit routing”. In: *Procs. of the Int. Symposium on Experimental Algorithms 2013*. Springer. 2013, pp. 43–54 (cit. on p. 234).

[176] E. W. Dijkstra. “A note on two problems in connexion with graphs”. In: *Numerische Mathematik* 1.1 (1959), pp. 269–271 (cit. on p. 233).

[177] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons. “A gentle introduction to imputation of missing values”. In: *Journal of Clinical Epidemiology* 59.10 (2006), pp. 1087–1091 (cit. on p. 13).

[178] Z. Dong, Y. Wu, M. Pei, and Y. Jia. “Vehicle Type Classification Using a Semisupervised Convolutional Neural Network”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.4 (2015), pp. 2247–2256 (cit. on p. 245).

[179] F. Doshi-Velez and B. Kim. “Towards a Rigorous Science of Interpretable Machine Learning”. In: *arXiv: Computing Research Repository* (2017). DOI: arXiv:1702.08608 (cit. on p. 16).

[180] K. Drew et al. “Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes”. In: *Molecular Systems Biology* 13.6 (2017), p. 932 (cit. on p. 86).

[181] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. “Fast approximation of matrix coherence and statistical leverage”. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 3475–3506 (cit. on p. 60).

[182] J. Du, C. Jiang, E. Gelenbe, L. Xu, J. Li, and Y. Ren. “Distributed data privacy preservation in IoT applications”. In: *IEEE Wireless Communications* 25.6 (2018), pp. 68–76 (cit. on p. 15).

[183] W. Du and Z. Zhan. “Using randomized response techniques for privacy-preserving data mining”. In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2003*. 2003, pp. 505–510 (cit. on p. 15).

[184] G. Duke, J. Briedis, and J. Green. “Survival of critically ill medical patients is time-critical”. In: *Critical Care and Resuscitation* 6.4 (2004) (cit. on p. 5).

[185] C. Dwork and A. Roth. “The Algorithmic Foundations of Differential Privacy”. In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407 (cit. on p. 232).

[186] J. F. Easton, C. R. Stephens, and M. Angelova. “Risk factors and prediction of very short term versus short/intermediate term post-stroke mortality: a data mining approach”. In: *Computers in Biology and Medicine* 54 (2014), pp. 199–210 (cit. on p. 15).

[187] C. Elkan. “Using the Triangle Inequality to Accelerate k-Means”. In: *Procs. of the Int. Conference on Machine Learning 2003*. 2003 (cit. on p. 151).

[188] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, M. Banyatsang, and O. Tabona. “A Survey On Missing Data in Machine Learning”. In: (2021) (cit. on p. 13).

[189] V. Emmerich, M. Döbele, T. Bauernhansl, D. Paulus-Rohmer, A. Schatz, and M. Weskamp. *Geschäftsmodell-Innovation durch Industrie 4.0-Chancen und Risiken für den Maschinen- und Anlagenbau*. 2015 (cit. on p. 104).

[190] H. J. Escalante et al. *Explainable and interpretable models in computer vision and machine learning*. Springer, 2018 (cit. on p. 17).

[191] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Procs. of the Int. Conference on Knowledge Discovery and Data Mining 1996*. AAAI Press, 1996, pp. 226–231 (cit. on p. 148).

[192] ETSI. *Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 2: Harmonised Std. covering the essential requirements of article 3.2 of Directive 2014/53/EU for non specific radio equipment*. 2016 (cit. on p. 315).

[193] *EU Science Hub Transport sector economic analysis*. https://ec.europa.eu/jrc/en/research-topic/transport-sector-economic-analysis. Accessed: 2021-02-01. 2021 (cit. on p. 213).

[194] R. Falkenberg, K. Heimann, and C. Wietfeld. “Discover Your Competition in LTE: Client-Based Passive Data Rate Prediction by Machine Learning”. In: *Procs. of the IEEE Global Communications Conference 2017*. Singapore, Dec. 2017, pp. 1–7 (cit. on p. 347). **SFB876-A4**

[195] R. Falkenberg, C. Ide, and C. Wietfeld. “Client-Based Control Channel Analysis for Connectivity Estimation in LTE Networks”. In: *Procs. of the IEEE Vehicular Technology Conference 2016*. Accepted for presentation. IEEE. Montréal, Canada, Sept. 2016 (cit. on pp. 344, 347, 348). **SFB876-A4**

[196] R. Falkenberg and C. Wietfeld. “FALCON: An accurate real-time monitor for client-based mobile network data analytics”. In: *Procs. of the IEEE Global Communications Conference 2019*. Waikoloa, Hawaii, USA: IEEE, Dec. 2019, pp. 1–7. URL: http://dx.doi.org/10.1109/GLOBECOM38437.2019.9014096 (cit. on pp. 345, 348, 351). **SFB876-A4**

[197] R. Falkenberg et al. “PhyNetLab: An IoT-based warehouse testbed”. In: *Procs. of the Federated Conference on Computer Science and Information Systems 2017*. Sept. 2017 (cit. on p. 200). **SFB876-A4**

[198] K. Fan, H. Zhang, S. Yan, L. Wang, W. Zhang, and J. Feng. “Learning a Generative Classifier from Label Proportions”. In: *Neurocomputing* 139 (Sept. 2014), pp. 47–55 (cit. on p. 142).

[199] L. Fan and G. Taylor. “Alter-CNN: An Approach to Learning from Label Proportions withApplication to Ice-Water Classification”. In: *Procs. of the Workshop on Learning and Privacy with Incomplete Data and Weak Supervision at the Conference on Neural Information Processing Systems 2015*. 2015 (cit. on p. 142).

[200] H. Fanaee-T and J. Gama. “Event labeling combining ensemble detectors and background knowledge”. In: *Progress in Artificial Intelligence* 2.2 (June 2014), pp. 113–127 (cit. on p. 164).

[201] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. “From Data Mining to Knowledge Discovery: An Overview”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. AAAI/MIT Press, 1996. Chap. 1, pp. 1–34 (cit. on p. 115).

[202] U. M. Fayyad and P. Smyth. “Towards Intelligent Trainable Tools for the Automated Analysis, Cataloging, and Searching of Digital Image Libraries: A Machine Learning Approach”. In: (1994) (cit. on p. 12).

[203] L. Feltrin, C. Buratti, E. Vinciarelli, R. D. Bonis, and R. Verdone. “LoRaWAN: Evaluation of Link- and System-Level Performance”. In: *IEEE Internet of Things Journal* 5.3 (June 2018), pp. 2249–2258 (cit. on p. 315).

[204] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli. *Power Electronics and Control Techniques for Maximum Energy Harvesting in Photovoltaic Systems*. CRC Press, 2012 (cit. on p. 198).

[205] F. T. Fernandes, T. A. de Oliveira, C. E. Teixeira, A. F. de Moraes Batista, G. Dalla Costa, and A. D. P. Chiavegatto Filho. “A multipurpose machine learning approach to predict COVID-19 negative prognosis in São Paulo, Brazil”. In: *Scientific Reports* 11.1 (2021), pp. 1–7 (cit. on p. 11).

[206] F. Finkeldey, A. Saadallah, P. Wiederkehr, and K. Morik. “Real-time prediction of process forces in milling operations using synchronized data fusion of simulation and sensor data”.

In: *Engineering Applications of Artificial Intelligence* Volume 94 (Sept. 2020). URL: https://www.sciencedirect.com/science/article/pii/S0952197620301627?dgcid=coauthor%5C#fig6 (cit. on pp. 159, 162, 166, 167, 171). **SFB876-B3**

[207] F. Finkeldey, A. Wirtz, T. Merhofe, and P. Wiederkehr. "Learning-Based Prediction of Pose-Dependent Dynamics". In: *Journal of Manufacturing and Materials Processing* 4.3 (2020). URL: https://www.mdpi.com/2504-4494/4/3/85 (cit. on pp. 159, 172–175). **SFB876-B3**

[208] C. Fischer. "Runtime and Accuracy Issues in Three-dimensional Finite Element Simulation of Machining". In: *International Journal of Machining and Machinability of Materials* 6.1 (2009), p. 35 (cit. on p. 158).

[209] B. Fish and L. Reyzin. "On the Complexity of Learning a Class Ratio from Unlabeled Data". In: *Journal of Artificial Intelligence Research* (Dec. 2020), pp. 1333–1349 (cit. on p. 139).

[210] J. Foley, A. van Dam, and S. Feiner. *Computer Graphics: Principles and Practice*. 2-nd edition in C. The systems programming series. Reading, Mass., Menlo Park, Calif., and New York: Addison-Wesley Publishing, 1997 (cit. on p. 159).

[211] V. Fortino et al. "Machine-learning–driven biomarker discovery for the discrimination between allergic and irritant contact dermatitis". In: *Procs. of the National Academy of Sciences* 117.52 (2020), pp. 33474–33485 (cit. on pp. 9, 10).

[212] S. Freitag, B. Cao, J. Ninic, and G. Meschke. "Surrogate modeling for mechanized tunneling simulations with uncertain data". In: *Procs. of the Int. Workshop on Reliable Engineering Computing 2014*. 2014, pp. 44–63 (cit. on p. 160).

[213] J. H. Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of Statistics* (2001), pp. 1189–1232 (cit. on p. 17).

[214] J. H. Friedman and B. E. Popescu. "Predictive learning via rule ensembles". In: *The Annals of Applied Statistics* 2.3 (2008), pp. 916–954 (cit. on p. 17).

[215] B. Friedrich. "The Effect of Autonomous Vehicles on Traffic". In: *Autonomous driving*. Berlin: Springer Open, 2016, pp. 317–334 (cit. on p. 295).

[216] J. Futoma, S. Hariharan, and K. Heller. "Learning to detect sepsis with a multitask Gaussian process RNN classifier". In: *Procs. of the Int. Conference on Machine Learning 2017*. PMLR. 2017, pp. 1174–1182 (cit. on p. 11).

[217] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich. "Traveling time prediction in scheduled transportation with journey segments". In: *Information Systems* (2015) (cit. on p. 235).

[218] J. Gama and M. M. Gaber. *Learning from data streams: processing techniques in sensor networks*. Springer, 2007 (cit. on p. 126).

[219] X. Gao, Z. Chen, S. Tang, Y. Zhang, and J. Li. "Adaptive weighted imbalance learning with application to abnormal activity recognition". In: *Neurocomputing* 173 (2016), pp. 1927–1935 (cit. on p. 125).

[220] M. R. Garey and D. S. Johnson. *Computers and Intractability – A Guide to NP-Completeness*. Int. Computer Science Series. New York: Freeman, 1979 (cit. on p. 88).

[221] V. N. Garla and C. Brandt. "Ontology-guided feature engineering for clinical text classification". In: *Journal of Biomedical Informatics* 45.5 (2012), pp. 992–998 (cit. on p. 12).

[222] E. Garrison and G. Marth. "Haplotype-based variant detection from short-read sequencing". In: *arXiv: Quantitative Biology* (July 2012). DOI: arXiv:1207.3907 (cit. on p. 55).

[223] R. Geisberger, P. Sanders, D. Schultes, and D. Delling. "Contraction hierarchies: Faster and simpler hierarchical routing in road networks". In: *Procs. of the Int. Workshop on Experimental and Efficient Algorithms 2008*. Springer. 2008, pp. 319–333 (cit. on p. 234).

[224] E. I. George and R. E. McCulloch. "Variable selection via Gibbs sampling". In: *Journal of the American Statistical Association* 88.423 (1993), pp. 881–889 (cit. on pp. 63, 64).

[225] J. George, L. Mary, and R. K. S. “Vehicle detection and classification from acoustic signal using ANN and KNN”. In: *Procs. of the Int. Conference on Control Communication and Computing 2013*. 2013, pp. 436–439 (cit. on p. 245).

[226] L. N. Geppert, K. Ickstadt, A. Munteanu, J. Quedenfeld, and C. Sohler. “Random projections for Bayesian regression”. In: *Statistics and Computing* 27.1 (2017), pp. 79–101. URL: http://doi.org/10.1007/s11222-015-9608-z (cit. on pp. 59, 60, 69). **SFB876-C4**

[227] L. N. Geppert, K. Ickstadt, A. Munteanu, and C. Sohler. “Streaming statistical models via Merge & Reduce”. In: *International Journal of Data Science and Analytics* 10.4 (2020), pp. 331–347. URL: https://doi.org/10.1007/s41060-020-00226-0 (cit. on pp. 59, 66, 68–70). **SFB876-C4**

[228] S. Gezici and H. V. Poor. “Position Estimation via Ultra-Wide-Band Signals”. In: *Procs. of the IEEE* 97.2 (Feb. 2009), pp. 386–403 (cit. on p. 181).

[229] Z. Gheid and Y. Challal. “Efficient and privacy-preserving k-means clustering for big data mining”. In: *Procs. of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications 2016*. IEEE. 2016, pp. 791–798 (cit. on p. 15).

[230] N. Ghosh et al. “Estimation of tool wear during CNC milling using neural network-based sensor fusion”. In: *Mechanical Systems and Signal Processing* 21.1 (2007), pp. 466–479 (cit. on p. 158).

[231] M. Goddard. “The EU General Data Protection Regulation (GDPR): European regulation that has a global impact”. In: *International Journal of Market Research* 59.6 (2017), pp. 703–705 (cit. on p. 231).

[232] M. Goerigk, M. Knoth, M. Müller-Hannemann, M. Schmidt, and A. Schöbel. “The Price of Robustness in Timetable Information”. In: *Procs. of the Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems 2011*. Ed. by A. Caprara and S. Kontogiannis. Vol. 20. OpenAccess Series in Informatics (OASIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011, pp. 76–87. DOI: 10.4230/OASIcs.ATMOS.2011.76. URL: http://drops.dagstuhl.de/opus/volltexte/2011/3268 (cit. on p. 234).

[233] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. “Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation”. In: *Journal of Computational and Graphical Statistics* 24.1 (2015), pp. 44–65 (cit. on p. 17).

[234] J. R. González, L. Armengol, X. Solé, E. Guinó, and V. Moreno. *SNPassoc: SNPs-Based Whole Genome Association Studies*. R package version 1.9-2. 2014. URL: https://CRAN.R-project.org/package=SNPassoc (cit. on p. 61).

[235] I. Goodfellow. “NIPS 2016 Tutorial: Generative Adversarial Networks”. In: *arXiv: Computing Research Repository* (2016). DOI: arXiv:1701.00160 (cit. on p. 16).

[236] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org (cit. on pp. 8, 245).

[237] I. Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27: Procs. of the 2014 Conference*. 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf (cit. on p. 38).

[238] N. Görnitz, M. Kloft, and U. Brefeld. “Active and Semi-Subpervised Data Domain Description”. In: *Procs. of the European Conference on Machine Learning and Knowledge Discovery in Databases 2009*. Ed. by W. Buntine, M. Grobelnik, D. Mladenic, and J. Shawe-Taylor. Berlin, Heidelberg: Springer, 2009, pp. 407–422 (cit. on p. 125).

[239] K. Gouda and M. Hassaan. “CSI_GED: An efficient approach for graph edit similarity computation”. In: *Procs. of the IEEE Int. Conference on Data Engineering 2016*. 2016, pp. 265–276 (cit. on p. 88).

[240] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour. “Artificial intelligence-based image classification for diagnosis of skin cancer: Challenges and opportunities”. In: *Computers in Biology and Medicine* (2020), p. 104065 (cit. on p. 17).

[241] E. Graf, C. Schmoor, W. Sauerbrei, and T. Schumacher. “Assessment and comparison of prognostic classification schemes for survival data”. In: *Statistics in Medicine* 18.17-18 (1999), pp. 2529–2545 (cit. on p. 65).

[242] B. Grüning et al. “Bioconda: sustainable and comprehensive software distribution for the life sciences”. In: *Nature Methods* 15.7 (July 2018), pp. 475–476 (cit. on p. 44).

[243] M. Guarnieri, S. Marinovic, and D. A. Basin. “Securing Databases from Probabilistic Inference”. In: *Procs. of the Computer Security Foundations Workshop 2017*. 2017, pp. 343–359 (cit. on p. 412).

[244] T. Gumbsch, C. Bock, M. Moor, B. Rieck, and K. Borgwardt. “Enhancing statistical power in temporal biomarker discovery through representative shapelet mining”. In: *Bioinformatics* 36.Supplement 2 (2020), pp. i840–i848 (cit. on p. 15).

[245] E. L. Gurevich, V. Temchura, K. Überla, and A. Zybin. “Analytical Features of Particle Counting Sensor based on Plasmon assisted Microscopy of Nano Objects”. In: *Sensors and Actuators B: Chemical* B 160.1 (Sept. 2011), pp. 1210–1215. URL: http://www.sciencedirect.com/science/article/pii/S0925400511008525 (cit. on pp. 24, 25). **SFB876-B2**

[246] L. Habel, C. Ide, C. Wietfeld, and M. Schreckenberg. “Improving Vehicular Traffic Simulations Using Real-Time Information on Environmental Conditions”. In: *Procs. of the IEEE Vehicular Technology Conference 2016*. IEEE. Montreal, Canada: IEEE, Sept. 2016. URL: http://dx.doi.org/10.1109/VTCFall.2016.7881044 (cit. on p. 288). **SFB876-B4**

[247] L. Habel, A. Molina, T. Zaksek, K. Kersting, and M. Schreckenberg. “Traffic simulations with empirical data – How to replace missing traffic flows?” In: *Traffic and Granular Flow ’15*. Ed. by V. L. Knoop and W. Daamen. Springer, May 2016, pp. 491–498. URL: http://dx.doi.org/10.1007/978-3-319-33482-0_62 (cit. on pp. 230, 289). **SFB876-B4**

[248] L. Habel and M. Schreckenberg. “Asymmetric Lane Change Rules for a Microscopic Highway Traffic Model”. In: *Procs. of Int. Conference on Cellular Automata for Research and Industry 2014*. Vol. 8751. LNCS. Krakow, Poland: Springer, Sept. 2014, pp. 620–629. URL: http://dx.doi.org/10.1007/978-3-319-11520-7_66 (cit. on p. 287). **SFB876-B4**

[249] F. Hadiji, A. Molina, S. Natarajan, and K. Kersting. “Poisson Dependency Networks: Gradient Boosted Models for Multivariate Count Data”. In: *Machine Learning Journal* 100.2 (2015), pp. 477–507. URL: http://dx.doi.org/10.1007/s10994-015-5506-z (cit. on pp. 230, 289). **SFB876-B4**

[250] S. Haendeler, A. Lewandowski, and C. Wietfeld. “Passive Detection of Wrong Way Drivers on Motorways Based on Low Power Wireless Communications”. In: *Procs. of the Vehicular Technology Conference 2014*. 2014, pp. 1–5 (cit. on pp. 242, 246, 247).

[251] M. Haferkamp, B. Sliwa, and C. Wietfeld. “A Low Cost Modular Radio Tomography System for Bicycle and Vehicle Detection and Classification”. In: *Procs. of the IEEE Int. Systems Conference 2021*. Vancouver, Canada, Apr. 2021 (cit. on pp. 243, 246). **SFB876-A4, SFB876-B4**

[252] M. Haferkamp et al. “Radio-based Traffic Flow Detection and Vehicle Classification for Future Smart Cities”. In: *Procs. of the IEEE Vehicular Technology Conference 2017 Workshop on Connecting All Things for Enabling Smart Cities (CONTEST)*. Sydney, June 2017 (cit. on pp. 243, 246). **SFB876-B4**

[253] S. Häger, S. Böcker, S. Jamali, T. Reinsch, and C. Wietfeld. “A Novel System Architecture for Small-Scale Motion Sensing Exploiting 5G mmWave Channels”. In: *Procs. of the IEEE Globecom Workshops 2021*. Madrid, Spain, Dec. 2021. DOI: 10.1109/GCWkshps52748.2021.9682166 (cit. on p. 388). **SFB876-A4**

[254] R. N. Hall. “Electron-Hole Recombination in Germanium”. In: 87.2 (1952), pp. 387–387. URL: https://link.aps.org/doi/10.1103/PhysRev.87.387 (cit. on p. 197).

[255] J. Y. Halpern and K. R. O’Neill. “Secrecy in Multiagent Systems”. In: *ACM Transactions on Information and System Security* 12.1 (2008) (cit. on p. 412).

[256] M. Hamer and R. D’Andrea. “Self-Calibrating Ultra-Wideband Network Supporting Multi-Robot Localization”. In: *IEEE Access* 6 (2018), pp. 22292–22304 (cit. on p. 190).

[257] F. Harary and R. Norman. “Some properties of line digraphs”. In: *Rendiconti del Circolo Matematico di Palermo* 9.2 (May 1960), pp. 161–168. URL: http://dx.doi.org/10.1007/BF02854581 (cit. on p. 237).

[258] P. E. Hart, N. J. Nilsson, and B. Raphael. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *Autonomous Mobile Robots*. Ed. by S. S. Iyengar and A. Elfes. (also to appear in IEEE Transactions on Systems Science, and Cybernetics, 1968). Los Alamitos, Calif. [u.a.]: IEEE Computer Soc. Press, 1991, pp. 375–382 (cit. on pp. 233, 235).

[259] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. 2nd. Statistics. Springer, 2009 (cit. on p. 146).

[260] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. New York, USA: Springer, 2001 (cit. on p. 124).

[261] Q. A. Hathaway et al. “Machine-learning to stratify diabetic patients using novel cardiac biomarkers and integrative genomics”. In: *Cardiovascular Diabetology* 18.1 (2019), pp. 1–16 (cit. on p. 10).

[262] M. Hayes. “Enables: Research Infrastructure Position Paper”. In: *Position Paper* 3.3 (2019), p. 58 (cit. on p. 262).

[263] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. “Mask R-CNN”. In: *arXiv: Computing Research Repository* (2017). DOI: arXiv:1703.06870 (cit. on p. 32).

[264] R. Hecht-Nielsen. “Theory of the Backpropagation Neural Network”. In: *Neural Networks for Perception (Vol. 2): Computation, Learning, Architectures*. USA: Harcourt Brace & Co., 1992, pp. 65–93 (cit. on p. 31).

[265] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. M. Kadie. “Dependency Networks for Collaborative Filtering and Data Visualization”. In: *Procs. of the Conference on Uncertainty in Artificial Intelligence 2000*. 2000, pp. 264–273. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1%5C&smnu=2%5C&article_id=31%5C&proceeding_id=16 (cit. on p. 230).

[266] K. Heimann, P. Gorczak, C. Bektas, F. Girke, and C. Wietfeld. “Software-Defined End-to-End Evaluation Platform for Quality of Service in Non-Standalone 5G Systems”. In: *Procs. of the IEEE Int. Systems Conference 2019*. IEEE, 2019 (cit. on pp. 384, 385). **SFB876-A4, SFB876-B4**

[267] K. Heimann, A. Marsch, B. Sliwa, and C. Wietfeld. “Reflecting Surfaces for Beyond Line-Of-Sight Coverage in Millimeter Wave Vehicular Networks”. In: *Procs. of the IEEE Vehicular Networking Conference 2020*. Dec. 2020. URL: https://ieeexplore.ieee.org/document/9318411 (cit. on pp. 385–387). **SFB876-A4, SFB876-B4**

[268] K. Heimann, J. Tiemann, S. Böcker, and C. Wietfeld. “Cross-Bearing based Positioning as a Feature of 5G Millimeter Wave Beam Alignment”. In: *Procs. of the IEEE Vehicular Technology Conference 2020*. Antwerp, Belgium, May 2020. URL: https://ieeexplore.ieee.org/document/9129182 (cit. on pp. 377, 381–383). **SFB876-B4**

[269] K. Heimann, J. Tiemann, S. Böcker, and C. Wietfeld. “On the Potential of 5G mmWave Pencil Beam Antennas for UAV Communications: An Experimental Evaluation”. In: *Procs. of the Int. ITG Workshop on Smart Antennas 2018*. Mar. 2018 (cit. on pp. 378, 379). **SFB876-A4, SFB876-B4**

[270] K. Heimann, J. Tiemann, D. Yolchyan, and C. Wietfeld. "Experimental 5G mmWave Beam Tracking Testbed for Evaluation of Vehicular Communications". In: *IEEE 5G World Forum* (Sept. 2019). URL: https://ieeexplore.ieee.org/document/8911692 (cit. on pp. 378, 380, 381). **SFB876-A4, SFB876-B4**

[271] D. Helbing. *Verkehrsdynamik: neue physikalische Modellierungskonzepte*. Springer-Verlag, 2013 (cit. on p. 226).

[272] C. Hernandez, C. Mella, G. Navarro, A. Olivera-Nappa, and J. Araya. "Protein complex prediction via dense subgraphs and false positive analysis". In: *PLoS One* 12.9 (2017), e0183460 (cit. on p. 86).

[273] J. Hernández-González, I. Inza, I. Granado, O. C. Basurko, J. A. Fernandes, and J. A. Lozano. "Aggregated outputs by linear models: An application on marine litter beaching prediction". In: *Information Sciences* 481 (2019), pp. 381–393 (cit. on p. 139).

[274] J. Hernández-González, I. Inza, and J. A. Lozano. "Learning Bayesian network classifiers from label proportions". In: *Pattern Recognition* 46.12 (2013), pp. 3425–3440. URL: http://www.sciencedirect.com/science/article/pii/S0031320313002069 (cit. on p. 142).

[275] S. Herranz et al. "The milling of airframe components with low rigidity: A general approach to avoid static and dynamic problems". In: *Procs. of the I MECH E Part B Journal of Engineering Manufacture* 219 (Nov. 2005), pp. 789–802 (cit. on p. 157).

[276] A. Higgins and E. Kozan. "Modeling train delays in urban networks". In: *Transportation Science* 32.4 (1998), pp. 346–357 (cit. on p. 234).

[277] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* (1997), pp. 1735–1780 (cit. on pp. 5, 11, 326).

[278] F. Holsboer. "How can we realize the promise of personalized antidepressant medicines?" In: *Nature Reviews Neuroscience* 9.8 (2008), pp. 638–646 (cit. on p. 3).

[279] M. ten Hompel, B. Vogel-Heuser, and T. Bauernhansl. *Handbuch Industrie 4.0*. Springer, 2020 (cit. on pp. 103, 110).

[280] S. Hoogendoorn, P. Bovy, and W. Daamen. *Pedestrian and evacuation dynamics*. 2002 (cit. on p. 227).

[281] G. Hooker. "Discovering additive structure in black box functions". In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2004*. 2004, pp. 575–580 (cit. on p. 17).

[282] R. P. Horgan and L. C. Kenny. "'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics". In: *The Obstetrician & Gynaecologist* 13.3 (2011), pp. 189–195 (cit. on p. 9).

[283] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*. Vol. 398. John Wiley & Sons, 2013 (cit. on p. 5).

[284] Q. Hou et al. "RankProd combined with genetic algorithm optimized artificial neural network establishes a diagnostic and prognostic prediction model that revealed C1QTNF3 as a biomarker for prostate cancer". In: *EBioMedicine* 32 (2018), pp. 234–244 (cit. on p. 10).

[285] R. Houari, A. Bounceur, A. K. Tari, and M. T. Kecha. "Handling missing data problems with sampling methods". In: *Procs. of the Int. Conference on Advanced Networking Distributed Systems and Applications 2014*. IEEE. 2014, pp. 99–104 (cit. on p. 13).

[286] C.-T. Hsueh, D. Liu, and H. Wang. "Novel biomarkers for diagnosis, prognosis, targeted therapy and clinical trials". In: *Biomarker Research* 1.1 (2013), pp. 1–2 (cit. on p. 9).

[287] M. Hu. "Cambridge Analytica's black box". In: *Big Data & Society* 7.2 (2020), p. 2053951720938091 (cit. on p. 14).

[288] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. "Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models". In: *Journal of Global Optimization* 34.3 (Mar. 2006), pp. 441–466 (cit. on p. 80).

[289] J. Huang et al. “An In-Depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance”. In: *Procs. of the ACM Conference on SIGCOMM 2013*. SIGCOMM ’13. New York, NY, USA: ACM, 2013, pp. 363–374 (cit. on p. 345).

[290] N. Hubballi, B. K. Patra, and S. Nandi. “NDoT: nearest neighbor distance based outlier detection technique”. In: *Procs. of the Int. Conference on Pattern Recognition and Machine Intelligence 2011*. Springer. 2011, pp. 36–42 (cit. on p. 14).

[291] T. Huck, A. Westenberger, M. Fritzsche, T. Schwarz, and K. Dietmayer. “Precise timestamping and temporal synchronization in multi-sensor fusion”. In: *Procs. of the IEEE Intelligent Vehicles Symposium 2011*. IEEE. 2011, pp. 242–247 (cit. on p. 120).

[292] U. Hunkeler, H. L. Truong, and A. Stanford-Clark. “MQTT-S – A publish/subscribe protocol for Wireless Sensor Networks”. In: *Procs. of the Int. Conference on Communication Systems Software and Middleware and Workshops 2008*. IEEE. 2008, pp. 791–798 (cit. on p. 220).

[293] S. L. Hyland et al. “Early prediction of circulatory failure in the intensive care unit using machine learning”. In: *Nature Medicine* 26.3 (2020), pp. 364–373 (cit. on pp. 5, 6, 11).

[294] R. Hyndman and Y. Khandakar. “Automatic Time Series Forecasting: The forecast Package for R”. In: *Journal of Statistical Software, Articles* 27.3 (2008), pp. 1–22. URL: https://www.jstatsoft.org/v027/i03 (cit. on p. 325).

[295] K. Ickstadt, M. Schäfer, and M. Zucknick. “Toward Integrative Bayesian Analysis in Molecular Biology”. In: *Annual Review of Statistics and Its Application* 5.1 (Mar. 2018), pp. 141–167. URL: https://doi.org/10.1146/annurev-statistics-031017-100438 (cit. on p. 63). **SFB876-C4**

[296] C. Ide. “Resource-efficient LTE machine-type communication in vehicular environments”. PhD thesis. TU Dortmund University, Mar. 2016 (cit. on pp. 333, 334). **SFB876-B4**

[297] C. Ide, R. Falkenberg, D. Kaulbars, and C. Wietfeld. “Empirical Analysis of the Impact of LTE Downlink Channel Indicators on the Uplink Connectivity”. In: *Procs. of the IEEE Vehicular Technology Conference 2016*. Accepted for presentation. IEEE. Nanjing, China, May 2016 (cit. on pp. 333, 345). **SFB876-A4, SFB876-B4**

[298] C. Ide et al. “LTE Connectivity and Vehicular Traffic Prediction based on Machine Learning Approaches”. In: *Procs. of the IEEE Vehicular Technology Conference 2015*. Boston, USA, Sept. 2015. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=%5C&arnumber=7391019%5C&url=http%5C%3A%5C%2F%5C%2Fieeexplore.ieee.org%5C%2Fxpls%5C%2Fabs_all.jsp%5C%3Farnumber%5C%3D7391019 (cit. on pp. 333, 334, 336). **SFB876-B4**

[299] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij. “A Survey of Incentive Techniques for Mobile Crowd Sensing”. In: *IEEE Internet of Things Journal* 2.5 (2015) (cit. on p. 298).

[300] A. Jain, M. Murty, and P. Flynn. “Data clustering: a review. ACM Computing Surveys”. In: 3.31 (1999), pp. 264–323 (cit. on p. 124).

[301] G. H. John and P. Langley. “Estimating Continuous Distributions in Bayesian Classifiers”. In: *Procs. of the Conference on Uncertainty in Artificial Intelligence 1995*. San Francisco: Morgan Kaufmann, 1995, pp. 338–345 (cit. on pp. 151, 152).

[302] J. W. Johnson. “Adapting Mask-RCNN for Automatic Nucleus Segmentation”. In: *arXiv: Computing Research Repository* (2018). DOI: arXiv:1805.00500 (cit. on p. 32).

[303] D. R. Jones, M. Schonlau, and W. J. Welch. “Efficient Global Optimization of Expensive Black-Box Functions”. In: *Journal of Global Optimization* 13.4 (1998), 455–492. URL: http://link.springer.com/article/10.1023/A:1008306431147 (cit. on p. 71).

[304] P. Jörke, S. Böcker, F. Liedmann, and C. Wietfeld. “Urban channel models for smart city IoT-networks based on empirical measurements of LoRa-lmks at 433 and 868 MHz”. In: *Procs. of the IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications 2017*. Vol. 2017-Octob. 2017 (cit. on pp. 315, 318). **SFB876-A4**

[305] P. Jörke, R. Falkenberg, and C. Wietfeld. “Power Consumption Analysis of NB-IoT and eMTC in Challenging Smart City Environments”. In: *Procs. of the IEEE Global Communications*

Conference Workshops, Workshop on Green and Sustainable 5G Wireless Networks 2018. IEEE, Dec. 2018 (cit. on pp. 264, 267, 268).

[306] P. Jörke, J. Güldenring, S. Böcker, and C. Wietfeld. "Coverage and Link Quality Improvement of Cellular IoT Networks with Multi-Operator and Multi-Link Strategies". In: *Procs. of the IEEE Vehicular Technology Conference 2019*. IEEE, Apr. 2019 (cit. on pp. 269, 270). **SFB876-A4**

[307] A. Juliani et al. "Unity: A General Platform for Intelligent Agents". In: *arXiv: Computing Research Repository* (2018). DOI: arXiv:1809.02627 (cit. on pp. 220–222).

[308] J. Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589 (cit. on p. 19).

[309] N. Kaempchen and K. Dietmayer. "Data synchronization strategies for multi-sensor fusion". In: *Procs. of the IEEE Conference on Intelligent Transportation Systems 2003*. Vol. 85. 1. 2003, pp. 1–9 (cit. on p. 120).

[310] H. Kagermann, W. Wahlster, and J. Helbig. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0*. 2013 (cit. on p. 104).

[311] V. Kann. "On the Approximability of the Maximum Common Subgraph Problem". In: *Procs. of the Annual Symposium on Theoretical Aspects of Computer Science 1992*. Vol. 577. LNCS. Springer, 1992, pp. 377–388 (cit. on p. 88).

[312] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. "Training Generative Adversarial Networks With Limited Data". In: *arXiv: Computing Research Repository* (2020). DOI: arXiv:2006.06676 (cit. on p. 39).

[313] M. W. Kattan and T. A. Gerds. "The index of prediction accuracy: an intuitive measure useful for evaluating risk prediction models". In: *Diagnostic and Prognostic Research* 2.1 (May 2018). URL: https://doi.org/10.1186/s41512-018-0029-2 (cit. on p. 73).

[314] G. Katz, E. C. R. Shin, and D. Song. "ExploreKit: Automatic Feature Generation and Selection". In: *Procs. of the IEEE Int. Conference on Data Mining 2016*. 2016 (cit. on p. 122).

[315] G. Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems 30: Procs. of the 2017 Conference*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf (cit. on pp. 5, 9).

[316] R. M. Keenan and L. .-. Tran. "Fall detection using Wi-Fi signals and threshold-based activity segmentation". In: *Procs. of the IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications 2020*. 2020, pp. 1–6 (cit. on p. 246).

[317] B. S. Kerner, S. L. Klenov, and M. Schreckenberg. "Simple cellular automaton model for traffic breakdown, highway capacity, and synchronized flow". In: *Physical Review E* 84.4 (2011), p. 46110 (cit. on p. 294).

[318] A. A. Khalil, A. J. Byrne, M. A. Rahman, and M. H. Manshaei. "REPlanner: Efficient UAV Trajectory-Planning using Economic Reinforcement Learning". In: *Procs. of the IEEE Int. Conference on Smart Computing 2021*. IEEE. 2021, pp. 153–160 (cit. on p. 222).

[319] A. Khan, Y. Wu, C. C. Aggarwal, and X. Yan. "NeMa: Fast Graph Search with Label Similarity". In: *Procs. of the Very Large Data Bases Endowment 2013* 6.3 (2013), pp. 181–192. URL: http://www.vldb.org/pvldb/vol6/p181-khan.pdf (cit. on p. 89).

[320] P. Kidger, J. Morrill, and T. Lyons. "Generalised Interpretable Shapelets for Irregular Time Series". In: *arXiv: Computing Research Repository* (2020). DOI: arXiv:2005.13948 (cit. on p. 9).

[321] O. Kienzle. "Die Bestimmung von Kräften und Leistungen an spanenden Werkzeugen und Werkzeugmaschinen". In: *VDI-Z* 94.11 (1952), pp. 299–305 (cit. on p. 160).

[322] H.-J. Kim, N.-O. Jo, and K.-S. Shin. "Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction". In: *Expert Systems with Applications* 59 (2016), pp. 226–234 (cit. on p. 125).

[323] J. Kim, D. Choi, and C. Li. “Inves: Incremental Partitioning-Based Verification for Graph Similarity Search”. In: *Advances in Database Technology - Procs. of the Int. Conference on Extending Database Technology 2019*. Ed. by M. Herschel, H. Galhardas, B. Reinwald, I. Fundulaki, C. Binnig, and Z. Kaoudi. OpenProceedings.org, 2019, pp. 229–240 (cit. on pp. 93, 94).

[324] D. Kinane et al. “Intelligent Synthesis and Real-time Response using Massive Streaming of Heterogeneous Data (INSIGHT) and its anticipated effect on Intelligent Transport Systems (ITS) in Dublin City, Ireland”. In: *Procs. of the ITS European Congress 2014*. 2014 (cit. on p. 238).

[325] A. K. Kiss, D. Hajdu, D. Bachrathy, and G. Stepan. “Operational stability prediction in milling based on impact tests”. In: *Mechanical Systems and Signal Processing* 103 (2018), pp. 327–339 (cit. on p. 173).

[326] J. P. Klein and M. L. Moeschberger. *Survival Analysis: Techniques for Censored and Truncated Data*. 2nd ed. Statistics for Biology and Health. Springer, 2003 (cit. on p. 79).

[327] L. A. Klein, M. K. Mills, and D. P. Gibson. “Traffic detector handbook: Third edition - volume I”. In: 2006 (cit. on p. 244).

[328] W. Knoll et al. “Surface Plasmon Fluorescence Techniques for Bioaffinity Studies.” In: *Handbook of Surface Plasmon Resonance*. Ed. by R. B. M. Schasfoort and A. J. Tudos. RSC Publishing, 2008, pp. 275–310 (cit. on p. 24).

[329] P. G. Knoops et al. “A machine learning framework for automated diagnosis and computer-assisted planning in plastic and reconstructive surgery”. In: *Scientific Reports* 9.1 (2019), pp. 1–12 (cit. on p. 6).

[330] E. M. Knorr, R. T. Ng, and V. Tucakov. “Distance-based outliers: algorithms and applications”. In: *The VLDB Journal – The International Journal on Very Large Data Bases* 8.3-4 (Feb. 2000), pp. 237–253 (cit. on p. 14).

[331] M. H. Ko, G. West, S. Venkatesh, and M. Kumar. “Using dynamic time warping for online temporal fusion in multisensor systems”. In: *Information Fusion* 9.3 (2008), pp. 370–388 (cit. on p. 166).

[332] M. F. Al-Kobaisi. “The Jawetz, Melnick & Adelberg’s Medical Microbiology 24th Edition”. In: *Sultan Qaboos University Medical Journal* 7.3 (Dec. 2007), pp. 273–275. URL: https://journals.squ.edu.om/index.php/squmj/article/view/1338 (cit. on p. 23).

[333] R. Kobayashi, Y. Mutuka, and T. Harada. “Risk Consistent Multi-Class Learning from Label Proportions”. In: *arXiv: Computing Research Repository* (Mar. 2022). DOI: arXiv:2203.12836 (cit. on p. 139).

[334] R. Kohavi, G. John, R. Long, and D. Manley. “MLC++: A Machine Learning Library in C++”. In: *Tools with Artificial Intelligence* (1994), pp. 740–743 (cit. on p. 169).

[335] G. Köksal, I. Batmaz, and M. C. Testik. “A review of data mining applications for quality improvement in manufacturing industry”. In: *Expert Systems with Applications* 38.10 (2011), pp. 13448–13467 (cit. on p. 115).

[336] R. I. Kondor and J. D. Lafferty. “Diffusion kernels on graphs and other discrete input spaces”. In: *Procs. of the Int. Conference on Machine Learning 2002*. 2002, pp. 315–322. URL: citeseer.ist.psu.edu/kondor02diffusion.html (cit. on p. 229).

[337] D. Kong, J. Zhu, C. Duan, L. Lu, and D. Chen. “Bayesian linear regression for surface roughness prediction”. In: *Mechanical Systems and Signal Processing* 142 (2020), p. 106770 (cit. on p. 158).

[338] H. J. de Koning et al. “Reduced lung-cancer mortality with volume CT screening in a randomized trial”. In: *New England Journal of Medicine* 382.6 (2020), pp. 503–513 (cit. on p. 7).

[339] B. Konrad, D. Lieber, and J. Deuse. “Striving for Zero Defect Production: Intelligent Manufacturing Control through Data Mining in Continuous Rolling Mill Processes”. In: *Robust*

Manufacturing Control: Procs. of the CIRP Sponsored Conference RoMaC 2012. Ed. by K. Windt. Lecture Notes in Production Engineering. CIRP. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 215–229. URL: http://link.springer.com/chapter/10.1007%5C%2F978-3-642-30749-2_16 (cit. on pp. 114, 116–118, 132). **SFB876-B3**

[340] R. P. H. Kooyman et al. *Handbook of Surface Plasmon Resonance*. Ed. by R. B. M. Schasfoort and A. J. Tudos. The Royal Society of Chemistry, 2008, P001–403. URL: http://dx.doi.org/10.1039/9781847558220 (cit. on p. 21).

[341] Y. Koren, R. Bell, and C. Volinsky. "Matrix factorization techniques for recommender systems". In: *Computer* 42.8 (2009), pp. 30–37 (cit. on p. 13).

[342] J. Köster, L. J. Dijkstra, T. Marschall, and A. Schönhuth. "Varlociraptor: enhancing sensitivity and controlling false discovery rate in somatic indel discovery". In: *Genome Biology* 21.1 (Apr. 2020), p. 98. URL: https://doi.org/10.1186/s13059-020-01993-6 (cit. on pp. 48–50, 53).

[343] J. Köster, F. Mölder, C. Schröder, T. Hartmann, J. Forster, and D. Laehnemann. *snakemake-workflows/dna-seq-varlociraptor: Version 2.0.5*. July 2021. URL: https://doi.org/10.5281/zenodo.5112918 (cit. on p. 54).

[344] J. Köster and S. Rahmann. "Massively parallel read mapping on GPUs with the q-group index and PEANUT". In: *PeerJ* 2 (2014), e606 (cit. on pp. 45, 57). **SFB876-C1**

[345] J. Köster and S. Rahmann. "Snakemake: a scalable bioinformatics workflow engine". In: *Bioinformatics* 28.19 (2012). PMID: 22908215, pp. 2520–2522. URL: http://dx.doi.org/10.1093/bioinformatics/bts480 (cit. on pp. 44, 95).

[346] H. Kotthaus et al. "RAMBO: Resource-Aware Model-Based Optimization with Scheduling for Heterogeneous Runtimes and a Comparison with Asynchronous Model-Based Optimization". In: *Procs. of the Int. Conference on Learning and Intelligent Optimization 2017*. 2017, pp. 180–195. URL: https://www.springerprofessional.de/en/rambo-resource-aware-model-based-optimization-with-scheduling-fo/15164982 (cit. on p. 77). **SFB876-A3**

[347] I. Koutsopoulos. "Optimal incentive-driven design of participatory sensing systems". In: *Procs. IEEE INFOCOM 2013*. 2013 (cit. on p. 298).

[348] D. Kozakov et al. "The ClusPro web server for protein-protein docking". In: *Nature Protocols* 12.2 (2017), pp. 255–278 (cit. on p. 86).

[349] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. "Recent Development and Applications of SUMO - Simulation of Urban MObility". In: *International Journal On Advances in Systems and Measurements* 5.3&4 (Dec. 2012), pp. 128–138 (cit. on p. 239).

[350] E. Kretschmann. "Determination of Optical Constants of Metals by Excitation of Surface Plasmons". In: *Zeitschrift für Physik* 241.4 (1971), pp. 313–324. URL: %3CGo%20to%20ISI%3E://WOS:A1971I477700003 (cit. on p. 24).

[351] N. Kriege, P. Giscard, and R. C. Wilson. "On Valid Optimal Assignment Kernels and Applications to Graph Classification". In: *arXiv: Computing Research Repository* (2016). DOI: arXiv:1606.01141 (cit. on p. 89). **SFB876-A6**

[352] N. M. Kriege, P.-L. Giscard, F. Bause, and R. C. Wilson. "Computing Optimal Assignments in Linear Time for Graph Matching". In: *arXiv: Computing Research Repository* arXiv:1901.10356 (2019) (cit. on p. 88). **SFB876-A6**

[353] N. M. Kriege, F. D. Johansson, and C. Morris. "A Survey on Graph Kernels". In: *Applied Network Science* 5.1 (2020), p. 6. URL: https://doi.org/10.1007/s41109-019-0195-3 (cit. on p. 87). **SFB876-A6**

[354] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn. "Digital Twin in manufacturing: A categorical literature review and classification". In: *IFAC-PapersOnLine* 51.11 (2018), pp. 1016–1022 (cit. on p. 108).

[355] A. Krogh and J. Vedelsby. “Neural Network Ensembles, Cross Validation, and Active Learning”. In: *Advances in Neural Information Processing Systems 7: Procs. of the 1994 Conference*. 1995 (cit. on p. 170).

[356] I. Krontiris and A. Albers. “Monetary incentives in participatory sensing using multi-attributive auctions”. In: *International Journal of Parallel, Emergent and Distributed Systems* 27.4 (2012) (cit. on pp. 298, 299).

[357] H. Kück and N. de Freitas. “Learning to Classify Individuals Based on Group Statistics”. In: *Procs. of the Conference in Uncertainty in Artificial Intelligence 2005*. 2005, pp. 332–339 (cit. on pp. 139, 140).

[358] A. Kulkarni, A. Seetharam, A. Ramesh, and J. Herath. “DeepChannel: Wireless Channel Quality Prediction Using Deep Learning”. In: *IEEE Transactions on Vehicular Technology* (2019), pp. 1–1 (cit. on p. 326).

[359] S. Kumar, E. Hamed, D. Katabi, and L. Erran Li. “LTE Radio Analytics Made Easy and Accessible”. In: *Procs. of the ACM Conference on SIGCOMM 2014*. New York, NY, USA: ACM, 2014, pp. 211–222 (cit. on pp. 345, 347).

[360] F. Kurtz, C. Bektas, N. Dorsch, and C. Wietfeld. “Network Slicing for Critical Communications in Shared 5G Infrastructures - An Empirical Evaluation”. In: *Procs. of the IEEE Int. Conference on Network Softwarization 2018*. IEEE. Canada, June 2018. URL: https://ieeexplore.ieee.org/document/8460110 (cit. on p. 360).

[361] A. Kuzmichev, J. Skolnik, A. Zybin, and R. Hergenroder. “Absolute analysis of nanoparticle suspension with surface plasmon microscopy”. In: *Analytical Chemistry* 90 (2018), pp. 10732–10737 (cit. on p. 26). **SFB876-B2**

[362] V. Lagani, G. Kortas, and I. Tsamardinos. “Biomarker signature identification in ’omics’ data with multi-class outcome”. In: *Computational and Structural Biotechnology Journal* 6.7 (2013), e201303004 (cit. on p. 9).

[363] P. Lakhani and B. Sundaram. “Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks”. In: *Radiology* 284.2 (2017), pp. 574–582 (cit. on p. 10).

[364] C. Lam, D. Yi, M. Guo, and T. Lindsey. “Automated detection of diabetic retinopathy using deep learning”. In: *AMIA Summits on Translational Science Proceedings* 2018 (2018), p. 147 (cit. on p. 6).

[365] J. Lamas, P.-M. Castro-Castro, A. Dapena, and F. Vazquez-Araujo. “Vehicle classification using the discrete fourier transform with traffic inductive sensors”. In: *Sensors* 15 (Oct. 2015) (cit. on p. 245).

[366] M. Lamraoui, M. Barakat, M. Thomas, and M. E. Badaoui. “Chatter detection in milling machines by neural network classification and feature selection”. In: *Journal of Vibration and Control* 21.7 (2015), pp. 1251–1266 (cit. on p. 160).

[367] M. Lang. “Automatische Modellselektion in der Überlebenszeitanalyse”. Dissertation. Dortmund: TU Dortmund University, 2015. URL: http://dx.doi.org/10.17877/DE290R-7245 (cit. on pp. 71, 75–77, 83).

[368] M. Lang, B. Bischl, and D. Surmann. “batchtools: Tools for R to Work on Batch Systems”. In: *The Journal of Open Source Software* 2.10 (2017) (cit. on p. 83).

[369] M. Lang, H. Kotthaus, P. Marwedel, C. Weihs, J. Rahnenführer, and B. Bischl. “Automatic Model Selection for High-Dimensional Survival Analysis”. In: *Journal of Statistical Computation and Simulation* 85.1 (2014), pp. 62–76. URL: http://www.tandfonline.com/eprint/fwxNXMbwDakhMqRrdMsJ/full (cit. on pp. 72, 75, 83). **SFB876-A3**

[370] M. Lang et al. “Mlr3: A Modern Object-Oriented Machine Learning Framework in R”. In: *Journal of Open Source Software* 4.44 (2019). 00000, p. 1903. URL: https://joss.theoj.org/papers/10.21105/joss.01903 (cit. on p. 83). **SFB876-A3**

[371] B. Langmead and S. L. Salzberg. “Fast gapped-read alignment with Bowtie 2”. In: *Nature Methods* 9.4 (2012), pp. 357–359. URL: http://www.nature.com/nmeth/journal/v9/n4/full/nmeth.1923.html (cit. on p. 45).

[372] G. Lanza et al. *Acatech Studie: Wandlungsfähige, menschzentrierte Strukturen in Fabriken und Netzwerken der Industrie 4.0*. 2018 (cit. on p. 104).

[373] K. Lasers. *Clubmax FB4 series Model Specific Operation Manual V271020*. Kvant Lasers. 2020 (cit. on p. 220).

[374] R. A. Laskowski, F. Gerick, and J. M. Thornton. “The structural basis of allosteric regulation in proteins”. In: *FEBS Letters* 583.11 (2009), pp. 1692–1698 (cit. on p. 95).

[375] F. Laue, P. Karunakaran, and R. Schober. “Detection Schemes and Model Mismatch Analysis for 5G Configured-Grant Access for URLLC”. In: *Procs. of the IEEE Globecom Workshops 2019*. Dec. 2019 (cit. on p. 365).

[376] S. Laur, H. Lipmaa, and T. Mielikäinen. “Cryptographically private support vector machines”. In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2006*. 2006, pp. 618–624 (cit. on p. 15).

[377] N. D. Lawrence. “Data Science and Digital Systems: The 3Ds of Machine Learning Systems Design”. In: *arXiv: Computing Research Repository* (2019). DOI: arXiv:1903.11241 (cit. on p. 107).

[378] A. Ledergerber, M. Hamer, and R. D’Andrea. “A robot self-localization system using one-way ultra-wideband communication”. In: *Procs. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems 2015*. 2015, pp. 3131–3137 (cit. on p. 190).

[379] H. K. Lee, R. Barlovic, M. Schreckenberg, and D. Kim. “Mechanical restriction versus human overreaction triggering congested traffic states”. In: *Physical Review Letters* 92.23 (2004), p. 238702 (cit. on pp. 286, 287, 290, 293, 294, 296).

[380] J.-S. Lee and B. Hoh. “Dynamic pricing incentive for participatory sensing”. In: *Pervasive and Mobile Computing* 6.6 (2010) (cit. on p. 298).

[381] J.-S. Lee and B. Hoh. “Sell your experiences: A market mechanism based incentive for participatory sensing”. In: *Procs. of IEEE PerCom 2010*. 2010 (cit. on pp. 298, 299).

[382] K. H. Lee, S. Chakraborty, and J. Sun. “Bayesian variable selection in semiparametric proportional hazards model for high dimensional survival data”. In: *International Journal of Biostatistics* 7.1 (2011) (cit. on pp. 62–64).

[383] S. Lee et al. “Robust Selection of Cancer Survival Signatures from High-Throughput Genomic Data Using Two-Fold Subsampling”. In: *PLoS ONE* 9 (2014), e108818 (cit. on pp. 72, 81). **SFB876-C1, SFB876-A3**

[384] B. Lela, D. Bajić, and S. Jozić. “Regression analysis, support vector machines, and Bayesian neural network approaches to modeling surface roughness in face milling”. In: *The International Journal of Advanced Manufacturing Technology* 42.11-12 (2009), pp. 1082–1088 (cit. on p. 158).

[385] J. E. Lenssen, V. Shpacovitch, D. Siedhoff, P. Libuschewski, R. Hergenröder, and F. Weichert. “A Review of Nano-Particle Analysis with the PAMONO-Sensor”. In: *Biosensors: Advances and Reviews* (2017), pp. 81–100 (cit. on pp. 21, 29–32, 35). **SFB876-B2**

[386] J. E. Lenssen, V. Shpacovitch, and F. Weichert. “Real-Time Virus Size Classification Using Surface Plasmon PAMONO Resonance and Convolutional Neural Networks”. In: *Bildverarbeitung für die Medizin 2017*. 2017, pp. 98–103. URL: http://dx.doi.org/10.1007/978-3-662-54345-0_26 (cit. on p. 37). **SFB876-B2**

[387] J. E. Lenssen et al. “Real-Time Low SNR Signal Processing for Nanoparticle Analysis with Deep Neural Networks”. In: *Procs. of the Int. Conference on Bio-inspired Systems and Signal Processing 2018*. (Best paper award). SciTePress, 2018. URL: https://doi.org/10.5220/0006596400360047 (cit. on pp. 30, 31). **SFB876-B2**

[388] J. Lerouge, Z. Abu-Aisheh, R. Raveaux, P. Héroux, and S. Adam. “New binary linear programming formulation to compute the graph edit distance”. In: *Pattern Recognition* 72 (2017), pp. 254–265 (cit. on pp. 88, 94, 95).

[389] M. Lesser. “Charge-coupled device (CCD) image sensors”. In: *High Performance Silicon Imaging*. Ed. by D. Durini. Second Edition. Woodhead Publishing Series in Electronic and Optical Materials. Woodhead Publishing, 2020 (cit. on p. 21).

[390] H. Li. “Minimap2: versatile pairwise alignment for nucleotide sequences”. In: *arXiv: Quantitative Biology* (Aug. 2017). DOI: arXiv:1708.01492 (cit. on p. 45).

[391] H. Li and R. Durbin. “Fast and accurate long-read alignment with Burrows-Wheeler transform”. In: *Bioinformatics* 26.5 (2010), pp. 589–595. URL: http : // bioinformatics . oxfordjournals.org/content/26/5/589 (cit. on pp. 45, 49).

[392] P. Li, L. He, X. Hu, Y. Zhang, L. Li, and X. Wu. “Concept based Short Text Stream Classification with Topic Drifting Detection”. In: *Procs. of the IEEE Int. Conference on Data Mining 2016*. 2016 (cit. on p. 345).

[393] S. C.-X. Li and B. Marlin. “A scalable end-to-end gaussian process adapter for irregularly sampled time series classification”. In: *Advances in Neural Information Processing Systems 29: Procs. of the 2016 Conference*. Vol. 29. 2016 (cit. on p. 13).

[394] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. “Federated learning: Challenges, methods, and future directions”. In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60 (cit. on p. 14).

[395] X. Li et al. “Automatic IMRT planning via static field fluence prediction (AIP-SFFP): a deep learning algorithm for real-time prostate treatment planning”. In: *Physics in Medicine & Biology* 65.17 (2020), p. 175014 (cit. on p. 8).

[396] Y. Li et al. “Multi-platform omics analysis reveals molecular signature for COVID-19 pathogenesis, prognosis and drug target discovery”. In: *Signal Transduction and Targeted Therapy* 6.1 (2021), pp. 1–11 (cit. on p. 11).

[397] Y. Liang and P. Zhao. “Similarity Search in Graph Databases: A Multi-Layered Indexing Approach”. In: *Procs. of the IEEE Int. Conference on Data Engineering 2017*. IEEE Computer Society, 2017, pp. 783–794 (cit. on pp. 93, 94).

[398] O. Liberg, M. Sundberg, E. Wang, J. Bergman, and J. Sachs. *Cellular Internet of Things: Technologies, Standards, and Performance*. Elsevier Science, 2017 (cit. on pp. 263, 264).

[399] P. Libuschewski. “Exploration of Cyber-Physical Systems for GPGPU Computer Vision-Based Detection of Biological Viruses”. PhD thesis. Dortmund, Germany: TU Dortmund University, 2017. URL: http://dx.doi.org/10.17877/DE290R-17952 (cit. on pp. 29, 30, 38). **SFB876-B2**

[400] P. Libuschewski, P. Marwedel, D. Siedhoff, and H. Müller. “Multi-Objective Energy-Aware GPGPU Design Space Exploration for Medical or Industrial Applications”. In: *Procs. of the Int. Conference on Signal-Image Technology and Internet-Based Systems 2014*. Ed. by I. C. Society. IEEE Computer Society, 2014, pp. 637–644. URL: http://dx.doi.org/10.1109/SITIS.2014.11 (cit. on p. 30). **SFB876-B2**

[401] P. Libuschewski, D. Siedhoff, C. Timm, A. Gelenberg, and F. Weichert. “Fuzzy-enhanced, Real-time capable Detection of Biological Viruses Using a Portable Biosensor”. In: *Procs. of the Int. Joint Conference on Biomedical Engineering Systems and Technologies 2013*. 2013, pp. 169–174 (cit. on pp. 29, 30). **SFB876-B2**

[402] P. Libuschewski, D. Siedhoff, C. Timm, and F. Weichert. “Mobile Detektion viraler Pathogene durch echtzeitfähige GPGPU-Fuzzy-Segmentierung”. In: *Bildverarbeitung für die Medizin 2013* (2013), pp. 326–331. URL: http://dx.doi.org/10.1007/978-3-642-36480-8_57 (cit. on p. 36). **SFB876-B2**

[403] P. Libuschewski et al. “Multi-Objective Computation Offloading for Mobile Biosensors via LTE”. In: *Procs. of the Int. Conference on Wireless Mobile Communication and Healthcare*

2014. 2014, pp. 226–229. URL: dx.doi.org/10.4108/icst.mobihealth.2014.257374 (cit. on pp. 35, 36). **SFB876-B2, SFB876-A4**

[404] D. Lieber. "Data Mining in der Qualitätslenkung am Beispiel der Stabstahlproduktion". PhD thesis. TU Dortmund University, 2018 (cit. on pp. 132, 133). **SFB876-B3**

[405] D. Lieber, B. Konrad, J. Deuse, M. Stolpe, and K. Morik. "Sustainable Interlinked Manufacturing Processes through Real-Time Quality Prediction". In: *Leveraging Technology for a Sustainable World: Procs. of the CIRP Conference on Life Cycle Engineering 2012*. Ed. by D. A. Dornfeld and B. S. Linke. CIRP. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 393–398. URL: http://link.springer.com/chapter/10.1007/978-3-642-29069-5_67 (cit. on pp. 116, 118, 119, 133). **SFB876-B3**

[406] T. Liebig, C. Körner, and M. May. "Fast Visual Trajectory Analysis Using Spatial Bayesian Networks". In: *Procs. of the IEEE International Conference on Data Mining Workshops 2009*. IEEE Computer Society, 2009, pp. 668–673. URL: http://doi.ieeecomputersociety.org/10.1109/ICDMW.2009.44 (cit. on p. 228).

[407] T. Liebig, C. Körner, and M. May. "Scalable Sparse Bayesian Network Learning for Spatial Applications". In: *Procs. of the IEEE International Conference on Data Mining Workshops 2008*. IEEE Computer Society, 2008, pp. 420–425. URL: http://dx.doi.org/10.1109/ICDMW.2008.124 (cit. on p. 228).

[408] T. Liebig, Z. Xu, and M. May. "Incorporating Mobility Patterns in Pedestrian Quantity Estimation and Sensor Placement". In: *Procs. of the Int. Workshop on Citizen Sensor Networks CitiSens 2012*. Ed. by J. Nin and D. Villatoro. Springer, 2013, pp. 67–80. URL: http://link.springer.com/content/pdf/10.1007%5C%2F978-3-642-36074-9_7 (cit. on p. 229).

[409] T. Liebig, Z. Xu, M. May, and S. Wrobel. "Pedestrian Quantity Estimation with Trajectory Patterns". In: *Procs. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2012*. Springer, 2012, pp. 629–643. URL: http://link.springer.com/chapter/10.1007%5C%2F978-3-642-33486-3_40 (cit. on p. 229).

[410] T. Liebig. "Privacy Preserving Centralized Counting of Moving Objects". In: *AGILE 2015*. Ed. by F. Bacao, M. Y. Santos, and M. Painho. Lecture Notes in Geoinformation and Cartography. Springer Int. Publishing, 2015, pp. 91–103. URL: http://dx.doi.org/10.1007/978-3-319-16787-9_6 (cit. on p. 239).

[411] T. Liebig, S. Peter, M. Grzenda, and K. Junosza-Szaniawski. "Dynamic Transfer Patterns for Fast Multi-modal Route Planning". In: *Societal Geo-innovation: Selected papers of the 20th AGILE Conference on Geographic Information Science*. Ed. by A. Bregt, T. Sarjakoski, R. van Lammeren, and F. Rip. Cham: Springer Int. Publishing, 2017, pp. 223–236. URL: http://dx.doi.org/10.1007/978-3-319-56759-4_13 (cit. on pp. 233, 235).

[412] T. Liebig and M. Sotzny. "On Avoiding Traffic Jams with Dynamic Self-Organizing Trip Planning". In: *Procs. of the Int. Conference on Spatial Information Theory 2017*. Ed. by E. Clementini, M. Donnelly, M. Yuan, C. Kray, P. Fogliaroni, and A. Ballatore. Vol. 86. Leibniz Int. Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 17:1–17:12. URL: http://drops.dagstuhl.de/opus/volltexte/2017/7761 (cit. on pp. 235, 240).

[413] J. Lin, E. Keogh, L. Wei, and S. Lonardi. "Experiencing SAX: a novel symbolic representation of time series". In: *Data Mining and Knowledge Discovery* 15.2 (2007), pp. 107–144 (cit. on p. 123).

[414] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. "Feature Pyramid Networks for Object Detection". In: *arXiv: Computing Research Repository* (2016). DOI: arXiv:1612.03144 (cit. on p. 32).

[415] T. Lin et al. "Microsoft COCO: Common Objects in Context". In: *arXiv: Computing Research Repository* (2014). DOI: arXiv:1405.0312 (cit. on pp. 8, 32).

[416] C. Lippert, J. Listgarten, Y. Liu, C. M. Kadie, R. I. Davidson, and D. Heckerman. “FaST linear mixed models for genome-wide association studies”. In: *Nature Methods* 8.10 (2011), pp. 833–835 (cit. on p. 9).

[417] B. Liu and H. Zhu. “Energy-effective data gathering for UAV-aided wireless sensor networks”. In: *Sensors* 19.11 (2019), p. 2506 (cit. on p. 213).

[418] S. Liu, X. Fang, Y. Guo, C. Liu, and J. Yang. “Theory and Realization of Secondary Task Assignment for Multi-UAV Pickup Based on Green Scheduling”. In: *Complexity* 2021 (2021) (cit. on p. 213).

[419] F. Llinares-López, L. Papaxanthos, D. Bodenham, D. Roqueiro, C. Investigators, and K. Borgwardt. “Genome-wide genetic heterogeneity discovery with categorical covariates”. In: *Bioinformatics* 33.12 (2017), pp. 1820–1828 (cit. on p. 9).

[420] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *Procs. of the IEEE Conference on Computer Vision and Pattern Recognition 2015*. 2015, pp. 3431–3440 (cit. on p. 176).

[421] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle. “The irace package: Iterated racing for automatic algorithm configuration”. In: *Operations Research Perspectives* 3 (2016), pp. 43–58. URL: https://doi.org/10.1016/j.orp.2016.09.002 (cit. on p. 75).

[422] D. Lymberopoulos and J. Liu. “The Microsoft Indoor Localization Competition: Experiences and Lessons Learned”. In: *IEEE Signal Processing Magazine* 34.5 (Sept. 2017), pp. 125–140 (cit. on p. 193).

[423] D. Lymberopoulos, J. Liu, et al. *Microsoft Indoor Localization 2018*. Apr. 2018. URL: https://www.microsoft.com/en-us/research/event/microsoft-indoor-localization-competition-ipsn-2018/ (cit. on p. 193).

[424] X. Ma and L. Gao. “Discovering protein complexes in protein interaction networks via exploring the weak ties effect”. In: *BMC Systems Biology* 6 Suppl 1 (2012), S6 (cit. on p. 86).

[425] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. “L-diversity: Privacy beyond k-anonymity”. In: *The ACM Transactions on Knowledge Discovery from Data* 1.1 (2007) (cit. on p. 15).

[426] J. MacQueen. “Some Methods for Classification and Analysis of Multivariate Observations”. In: *Procs. of the Berkeley Symposium on Mathematical Statistics and Probability 1967*. Ed. by L. M. L. Cam and J. Neyman. Vol. 1. University of California Press, 1967, pp. 281–297 (cit. on pp. 124, 148).

[427] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata. “Road damage detection using deep neural networks with images captured through a smartphone”. In: *Computer Aided Civil and Infrastructure Engineering* 33.2 (2018), pp. 1–15 (cit. on p. 309).

[428] M. Mahdavi et al. “A machine learning based exploration of COVID-19 mortality risk”. In: *Plos one* 16.7 (2021), e0252384 (cit. on pp. 10, 11).

[429] N. H. Mahmood et al. “Machine type communications: key drivers and enablers towards the 6G era”. In: *EURASIP Journal on Wireless Communications and Networking* 2021.1 (June 2021), p. 134. URL: https://doi.org/10.1186/s13638-021-02010-5 (cit. on pp. 329, 330).
SFB876-A4

[430] P.-F. Marteau and S. Gibet. “On recursive edit distance kernels with application to time series classification”. In: *IEEE transactions on neural networks and learning systems* 26.6 (2014), pp. 1121–1133 (cit. on pp. 9, 11).

[431] M. Martin. “Cutadapt removes adapter sequences from high-throughput sequencing reads”. In: *EMBnet.journal* 17.1 (2011) (cit. on p. 55).

[432] M. Masoudinejad, J. Emmerich, D. Kossmann, A. Riesner, M. Roidl, and M. ten Hompel. “Development of a measurement platform for indoor photovoltaic energy harvesting in materials

handling applications". In: *Procs. of the Int. Renewable Energy Congress 2015*. Mar. 2015, pp. 1–6. URL: http://www.sciencedirect.com/science/article/pii/S2210670715300329 (cit. on p. 203). **SFB876-A4**

[433] M. Masoudinejad. "Data-Sets for Indoor Photovoltaic Behavior in Low Lighting Conditions". In: 5.2 (2020), p. 32. URL: https://www.mdpi.com/2306-5729/5/2/32 (cit. on pp. 203, 204). **SFB876-A4**

[434] M. Masoudinejad. "Modeling Energy Supply Unit of Ultra-Low Power Devices with Indoor Photovoltaic Harvesting". PhD Thesis. TU Dortmund University, 2020. URL: https://eldorado.tu-dortmund.de/handle/2003/39765 (cit. on pp. 195, 196, 200). **SFB876-A4**

[435] M. Masoudinejad, J. Emmerich, D. Kossmann, A. Riesner, M. Roidl, and M. ten Hompel. "A Measurement Platform for Photovoltaic Performance Analysis in Environments with Ultra-Low Energy Harvesting Potential". In: *Sustainable Cities and Society* 25 (2016), pp. 74–81 (cit. on p. 203). **SFB876-A4**

[436] F.-I. für Materialfluss und Logistik IML Dortmund. "Logistik entdecken, Magazin des Fraunhofer-Instituts für Materialfluss und Logistik IML Dortmund". In: *Logistik entdecken* 2019.19 (2019) (cit. on p. 220).

[437] F. Mattern and C. Floerkemeier. "From the Internet of Computers to the Internet of Things". In: *From Active Data Management to Event-Based Systems and More*. Springer, 2010, pp. 242–259 (cit. on p. 105).

[438] M. May. *Eight ways machine learning is assisting medicine*. 2021 (cit. on p. 4).

[439] J. D. Mazimpaka and S. Timpf. "A visual and computational analysis approach for exploring significant locations and time periods along a bus route". In: *Procs. of the ACM SIGSPATIAL Int. Workshop on Computational Transportation Science 2016*. ACM. 2016, pp. 43–48 (cit. on pp. 233, 234).

[440] W. McLaren et al. "The Ensembl Variant Effect Predictor". In: *Genome Biology* 17.1 (June 2016), p. 122. URL: https://doi.org/10.1186/s13059-016-0974-4 (cit. on p. 56).

[441] M. J. Mehlman. "Medical practice guidelines as malpractice safe harbors: illusion or deceit?" In: *Journal of Law, Medicine & Ethics* 40.2 (2012), pp. 286–300 (cit. on p. 19).

[442] P. Melville. "Creating Diverse Ensemble Classifiers to Reduce Supervision". PhD thesis. Austin, TX, USA, 2005 (cit. on p. 163).

[443] G. Mendel. "Versuche über Pflanzen-Hybriden". In: 4 (1866). Place: BrünnSeperatabdruck aus dem IV. Bande der Verhandlungen des naturforschenden Vereines; mit handschriftlichen Korrekturen Mendels., pp. 3–47. URL: http://vlp.mpiwg-berlin.mpg.de/references?id=lit26745 (cit. on pp. 48, 51).

[444] I. Mierswa. "Non-Convex and Multi-Objective Optimization in Data Mining". PhD thesis. TU Dortmund University, 2008 (cit. on p. 147).

[445] I. Mierswa and K. Morik. "Automatic Feature Extraction for Classifying Audio Data". In: *Machine Learning Journal* 58 (2005), pp. 127–149 (cit. on pp. 119, 122, 123).

[446] Mikhaylov, Konstantin et al. "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology". In: *European Wireless 2016* (2016), pp. 119–124 (cit. on p. 315).

[447] T. M. Mitchell. *Machine Learning*. New York: McGraw Hill, 1997 (cit. on pp. 139, 142).

[448] J. Mitra et al. "Statistical machine learning to identify traumatic brain injury (TBI) from structural disconnections of white matter networks". In: *NeuroImage* 129 (2016), pp. 247–259 (cit. on p. 10).

[449] H.-C. Möhring, P. Wiederkehr, K. Erkorkmaz, and Y. Kakinuma. "Self-optimizing machining systems". In: *CIRP Annals* 69.2 (2020), pp. 740–763 (cit. on p. 158).

[450] F. Mölder et al. "Sustainable data analysis with Snakemake". In: *F1000Research* 10 (Jan. 2021), p. 33. URL: https://f1000research.com/articles/10-33/v1 (cit. on p. 44). **SFB876-C1**

[451] A. Molina, A. Munteanu, and K. Kersting. “Core Dependency Networks”. In: *Procs. of the AAAI Conference on Artificial Intelligence 2018*. 2018. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16847 (cit. on p. 230). **SFB876-B4, SFB876-C4**

[452] A. Molina, S. Natarajan, and K. Kersting. “Poisson Sum-Product Networks: A Deep Architecture for Tractable Multivariate Poisson Distributions”. In: *Procs. of the AAAI Conference on Artificial Intelligence 2017*. 2017, pp. 2357–2363 (cit. on p. 230). **SFB876-B4, SFB876-A6**

[453] C. Molnar, G. Casalicchio, and B. Bischl. “Interpretable machine learning–a brief history, state-of-the-art and challenges”. In: *Procs. of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2020*. Springer. 2020, pp. 417–431 (cit. on p. 17).

[454] M. Moor, M. Horn, B. Rieck, D. Roqueiro, and K. Borgwardt. “Early recognition of sepsis with Gaussian process temporal convolutional networks and dynamic time warping”. In: *Procs. of the Machine Learning for Healthcare Conference 2019*. PMLR. 2019, pp. 2–26 (cit. on pp. 10, 11).

[455] M. Moor, B. Rieck, M. Horn, C. R. Jutzeler, and K. Borgwardt. “Early prediction of sepsis in the ICU using machine learning: a systematic review”. In: *Frontiers in medicine* 8 (2021), p. 348 (cit. on p. 11).

[456] E. Moradi, A. Pepe, C. Gaser, H. Huttunen, J. Tohka, A. D. N. Initiative, et al. “Machine learning framework for early MRI-based Alzheimer’s conversion prediction in MCI subjects”. In: *Neuroimage* 104 (2015), pp. 398–412 (cit. on pp. 9, 10).

[457] M. Morenza-Cinos, V. Casamayor-Pujol, J. Soler-Busquets, J. L. Sanz, R. Guzmán, and R. Pous. “Development of an RFID inventory robot (AdvanRobot)”. In: *Robot Operating System (ROS)*. Springer, 2017, pp. 387–417 (cit. on p. 217).

[458] K. Morik and S. Wessel. “Incremental Signal to Symbol Processing”. In: *Making Robots Smarter – Combining Sensing and Action through Robot Learning*. Ed. by K. Morik, V. Klingspor, and M. Kaiser. Kluwer Academic Publ., 1999. Chap. 11, pp. 185–198 (cit. on p. 123).

[459] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas. “Exemplar queries: a new way of searching”. In: *VLDB – The International Journal on Very Large Data Bases* 25.6 (2016), pp. 741–765. URL: https://doi.org/10.1007/s00778-016-0429-2 (cit. on p. 89).

[460] M. Moya, M. Koch, and L. Hostetler. “One-class classifier networks for target recognition applications.” In: *Procs. of the World Congress on Neural Networks 1993*. Int. Neural Network Society, 1993, pp. 797–801 (cit. on p. 125).

[461] M. Müller. “Dynamic time warping”. In: *Information Retrieval for Music and Motion* (2007), pp. 69–84 (cit. on p. 121).

[462] M. Müller-Hannemann and M. Schnee. “Efficient timetable information in the presence of delays”. In: *Robust and Online Large-Scale Optimization*. Springer, 2009, pp. 249–272 (cit. on p. 234).

[463] A. Munteanu, C. Schwiegelshohn, C. Sohler, and D. P. Woodruff. “On Coresets for Logistic Regression”. In: *Advances in Neural Information Processing Systems 31: Procs. of the 2018 COnference*. 2018. URL: http://papers.nips.cc/paper/7891-on-coresets-for-logistic-regression (cit. on p. 59). **SFB876-A2, SFB876-C4**

[464] Y. Murakami, T. Magome, K. Matsumoto, T. Sato, Y. Yoshioka, and M. Oguchi. “Fully automated dose prediction using generative adversarial networks in prostate cancer patients”. In: *PloS one* 15.5 (2020), e0232697 (cit. on p. 6).

[465] A. Murrenhoff, M. Roidl, and M. ten Hompel. “Control concept for ü r virtualized and learnable ä capable material flow systems”. In: *Logistics Journal: Proceedings* 2019.12 (2019) (cit. on pp. 219, 220).

[466] D. R. Musicant, J. M. Christensen, and J. F. Olson. “Supervised Learning by Training on Aggregate Outputs”. In: *Procs. of the IEEE Int. Conference on Data Mining 2007*. Washington, DC, USA: IEEE Computer Society, Oct. 2007, pp. 252–261 (cit. on pp. 137, 139, 141).

[467] K. Nagel and M. Schreckenberg. “A cellular automaton model for freeway traffic”. In: *Journal de Physique 1* 2 (1992), pp. 2221–2229 (cit. on pp. 285, 287).

[468] F. Naumann, A. Bilke, J. Bleiholder, and M. Weis. “Data fusion in three steps: Resolving inconsistencies at schema-, tuple-, and value-level”. In: *IEEE Data Engineering Bulletin* 29.2 (2006), pp. 21–31 (cit. on pp. 166, 167).

[469] A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera. “Handling incomplete heterogeneous data using vaes”. In: *Pattern Recognition* 107 (2020), p. 107501 (cit. on p. 13).

[470] R. A. Nazib and S. Moh. “Routing protocols for unmanned aerial vehicle-aided vehicular ad hoc networks: A survey”. In: *IEEE Access* 8 (2020), pp. 77535–77560 (cit. on p. 274).

[471] T. R. Network. “The Cancer Genome Atlas Program”. In: (2021). URL: http://cancergenome.nih.gov/ (cit. on p. 64).

[472] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich. “Better approach to mobile ad-hoc networking (BATMAN)”. In: *IETF draft* (2008), pp. 1–24 (cit. on p. 273).

[473] J. Neumann, A. W. Burks, et al. *Theory of self-reproducing automata*. Vol. 1102024. University of Illinois press Urbana, 1966 (cit. on p. 227).

[474] K. Ni et al. “Sensor network data fault types”. In: *ACM Transactions on Sensor Networks (TOSN)* 5.3 (2009), pp. 1–29 (cit. on p. 119).

[475] T. L. and Nico Piatkowski and Christian Bockermann and Katharina Morik. “Predictive Trip Planning - Smart Routing in Smart Cities”. In: *Procs. of the Joint EDBT/ICDT Workshops 2014*. Vol. 1133. CEUR-WS.org, 2014, pp. 331–338 (cit. on p. 230). **SFB876-A1**

[476] B. Niehoefer, F. Schweikowski, S. Lehnhausen, and C. Wietfeld. “Cloud-aided SDR Solution for Lane-Specific Vehicle Positioning via Local Interference Compensation”. In: *Procs. of IEEE Aerospace Conference 2014*. Big Sky, USA, Mar. 2014 (cit. on p. 242).

[477] B. Niehöfer, F. Schweikowski, and C. Wietfeld. “LOcal interferenCe compensATion (LOCATe) for GNSS-based Lane-Specific Positioning of Vehicles”. In: *Procs. of the IEEE Vehicular Technology Conference 2016*. Nanjing, China, May 2016 (cit. on p. 242).

[478] F. Niemann et al. “Lara: Creating a dataset for human activity recognition in logistics using semantic attributes”. In: *Sensors* 20.15 (2020), p. 4083 (cit. on p. 218).

[479] NREL. *Reference Air Mass 1.5 Spectra*. 2012 (cit. on p. 201).

[480] K. O’Shea and R. Nash. “An introduction to convolutional neural networks”. In: *arXiv: Computing Research Repository* (2015). DOI: arXiv:1511.08458 (cit. on p. 8).

[481] G. Oddi, D. Macone, A. Pietrabissa, and F. Liberati. “A proactive link-failure resilient routing protocol for MANETs based on reinforcement learning”. In: *Procs. of the Mediterranean Conference on Control Automation 2012*. 2012, pp. 1259–1264 (cit. on pp. 274, 275).

[482] I. Oleaga, C. Pardo, J. J. Zulaika, and A. Bustillo. “A machine-learning based solution for chatter prediction in heavy-duty milling machines”. In: *Measurement* 128 (2018), pp. 34–44 (cit. on p. 158).

[483] R. Olfati-Saber. “Flocking for multi-agent dynamic systems: Algorithms and theory”. In: *IEEE Transactions on automatic control* 51.3 (2006), pp. 401–420 (cit. on p. 215).

[484] D. L. Olson and D. Delen. *Advanced Data Mining Techniques*. Springer-Verlag Berlin Heidelberg, 2008 (cit. on p. 115).

[485] S. Oota. “Somatic mutations – Evolution within the individual”. In: *Methods*. RNA-Seq: Methods and Applications 176 (Apr. 2020), pp. 91–98. URL: https://www.sciencedirect.com/science/article/pii/S1046202318303827 (cit. on p. 53).

[486] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain. “Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives”. In: *IEEE Access* 7 (2019), pp. 81057–81105 (cit. on p. 274).

[487] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. R. Acharya. “Automated detection of COVID-19 cases using deep neural networks with X-ray images”. In: *Computers in biology and medicine* 121 (2020), p. 103792 (cit. on p. 6).

[488] S. L. Pardau. “The California consumer privacy act: towards a European-style privacy regime in the United States”. In: *Journal of Technology Law & Policy* 23 (2018), p. 68 (cit. on p. 14).

[489] R. B. Parikh, S. Teeple, and A. S. Navathe. “Addressing bias in artificial intelligence in health care”. In: *The Journal of the American Medical Association (JAMA)* 322.24 (2019), pp. 2377–2378 (cit. on p. 17).

[490] H. Park, H. Lee, and C. Seok. “High-resolution protein-protein docking by global optimization: recent advances and future challenges”. In: *Current Opinion in Structural Biology (COSB)* 35 (2015), pp. 24–31 (cit. on p. 86).

[491] J. Park, K. Law, R. Bhinge, M. Chen, D. Dornfeld, and S. Rachuri. “Realtime energy prediction for a milling machine tool using sparse Gaussian process regression”. In: Oct. 2015, pp. 1451–1460 (cit. on p. 158).

[492] K. Parry, L. N. Geppert, A. Munteanu, and K. Ickstadt. “Cross-Leverage Scores for Selecting Subsets of Explanatory Variables”. In: *arXiv: Statistics* (2021). DOI: arXiv:2109.08399 (cit. on pp. 60, 61). **SFB876-C4**

[493] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai. “A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions”. In: *IEEE Communications Surveys Tutorials* 20.4 (May 2018), pp. 3098–3130 (cit. on p. 364).

[494] G. Patrini, R. Nock, T. Caetano, and P. Rivera. “(Almost) No Label No Cry”. In: *NIPS 27*. Curran Associates, Inc., 2014, pp. 190–198 (cit. on pp. 140, 141).

[495] D. Paul, E. Bair, T. Hastie, and R. Tibshirani. “Preconditioning for feature selection and regression in high-dimensional problems”. In: *The Annals of Statistics* 36.4 (Aug. 2008), pp. 1595–1618 (cit. on p. 81).

[496] W. R. Pearson. “Selecting the Right Similarity-Scoring Matrix”. In: *Current Protocols in Bioinformatics* 43 (2013), pp. 1–9 (cit. on p. 86).

[497] M. Pellegrini, M. Baglioni, and F. Geraci. “Protein complex prediction for large protein protein interaction networks with the Core&Peel method”. In: *BMC Bioinformatics* 17.Suppl 12 (2016), p. 372 (cit. on p. 86).

[498] B. Peng, T. Bergs, D. Schraknepper, F. Klocke, and B. Döbbeler. “A hybrid approach using machine learning to predict the cutting forces under consideration of the tool wear”. In: *Procedia CIRP* 82 (2019), pp. 302–307 (cit. on p. 158).

[499] N. Piatkowski, S. Lee, and K. Morik. “Spatio-temporal random fields: compressible representation and distributed estimation”. In: *Machine Learning* 93.1 (2013), pp. 115–139. URL: http://link.springer.com/article/10.1007%5C%2Fs10994-013-5399-7 (cit. on p. 230). **SFB876-A1**

[500] *Picard toolkit*. http://broadinstitute.github.io/picard/. 2019 (cit. on p. 55).

[501] B. Pinkas. “Cryptographic techniques for privacy-preserving data mining”. In: *ACM SIGKDD Explorations Newsletter* 4.2 (2002), pp. 12–19 (cit. on p. 15).

[502] I. Pires, N. Garcia, N. Pombo, and F. Flórez-Revuelta. “From data acquisition to data fusion: a comprehensive review and a roadmap for the identification of activities of daily living using mobile devices”. In: *Sensors* 16.2 (2016), p. 184 (cit. on p. 166).

[503] D. Plakhotnik et al. “Integration of Process Monitoring Data into CAM Simulation.” In: (2018). IVth Int. Conference on High Speed Machining, Donostia/San Sebastian - Spain, 17-18 April 2018 (cit. on p. 158).

[504] A. van der Pol E. andBoing, P. Harrison, A. Sturk, and R. Nieuwland. “Classification, Functions, and Clinical Relevance of Extracellular Vesicles”. In: *Pharmacological Reviews* 64 (2012), pp. 676–705 (cit. on p. 23).

[505] E. van der Pol, A. N. Boing, E. L. Gool, and R. Nieuwland. “Recent developments in the nomenclature, presence, isolation, detection and clinical impact of extracellular vesicles”. In: *Journal of Thrombosis and Haemostasis* 14.1 (2016), pp. 48–56. URL: %3CGo%20to%20ISI%3E://WOS:000370661100005 (cit. on p. 23).

[506] H. Poon and P. Domingos. “Sum-product networks: A new deep architecture”. In: *Procs. of the IEEE Int. ConferenceComputer Vision Workshops 2011*. IEEE. 2011, pp. 689–690 (cit. on p. 230).

[507] F. Potorti et al. “Comparing the Performance of Indoor Localization Systems through the EvAAL Framework”. In: *Sensors* 17.10 (2017) (cit. on p. 192).

[508] A. Pottmeier, C. Thiemann, A. Schadschneider, and M. Schreckenberg. “Mechanical Restriction Versus Human Overreaction: Accident Avoidance and Two-Lane Traffic Simulations”. In: *Traffic And Granular Flow*. Ed. by Springer. Springer, 2005, pp. 503–508 (cit. on p. 293).

[509] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian. “Crazyswarm: A large nano-quadcopter swarm”. In: *Procs. of the IEEE Int. Conference on Robotics and Automation 2017*. IEEE. 2017, pp. 3299–3304 (cit. on pp. 216, 217, 221).

[510] W. N. Price, S. Gerke, and I. G. Cohen. “Potential liability for physicians using artificial intelligence”. In: *JAMA* 322.18 (2019), pp. 1765–1766 (cit. on p. 19).

[511] N. C. Purandare and V. Rangarajan. “Imaging of lung cancer: implications on staging and management”. In: *The Indian journal of radiology & imaging* 25.2 (2015), p. 109 (cit. on p. 8).

[512] T. G. Purdie, R. E. Dinniwell, D. Letourneau, C. Hill, and M. B. Sharpe. “Automated planning of tangential breast intensity-modulated radiotherapy using heuristic optimization”. In: *International Journal of Radiation Oncology* Biology* Physics* 81.2 (2011), pp. 575–583 (cit. on p. 8).

[513] Y. Qian, Q. Tong, and B. Wang. “Multi-Class Learning from Label Proportions for Bank Customer Classification”. In: *Procedia Computer Science* 162 (2019). Procs. of the Int. Conference on Information Technology and Quantitative Management 2019, pp. 421–428 (cit. on p. 139).

[514] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. “Estimating Labels from Label Proportions”. In: *Journal of Machine Learning Research* 10 (Dec. 2009), pp. 2349–2374 (cit. on pp. 139, 151).

[515] J. Quedenfeld and S. Rahmann. “Variant tolerant read mapping using min-hashing”. In: *arXiv: Quantitative Biology* (Feb. 2017). DOI: arXiv:1702.01703 (cit. on p. 47).

[516] J. R. Quinlan. “Induction of Decision Trees”. In: *Machine Learning* 1.1 (Mar. 1986), pp. 81–106. URL: http://portal.acm.org/citation.cfm?id=637962.637969 (cit. on p. 152).

[517] S. Rahmann, M. Martin, J. H. Schulte, J. Köster, T. Marschall, and A. Schramm. “Identifying Transcriptional miRNA Biomarkers by Integrating High-Throughput Sequencing and Real-Time PCR Data”. In: *Methods* 59.1 (Jan. 2013), pp. 154–163 (cit. on p. 45). **SFB876-C1, SFB876-B1**

[518] G. Raja, S. Anbalagan, V. S. Narayanan, S. Jayaram, and A. Ganapathisubramaniyan. “Inter-UAV collision avoidance using Deep-Q-learning in flocking environment”. In: *Procs. of the IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference 2019*. IEEE. 2019, pp. 1089–1095 (cit. on p. 222).

[519] S. Rajab, M. O. Al Kalaa, and H. Refai. “Classification and speed estimation of vehicles via tire detection using single-element piezoelectric sensor”. In: *Journal of Advanced Transportation* 50.7 (2016), pp. 1366–1385 (cit. on p. 245).

[520] A. K. Ramachandran Venkatapathy, H. Bayhan, F. Zeidler, and M. ten Hompel. “Human Machine Synergies in Intra-Logistics: Creating a Hybrid Network for Research and Technologies”. In: *Procs. of the Workshop on Information Technologies for Logistics 2017* (2017) (cit. on p. 218). **SFB876-A4**

[521] B. Rao, A. G. Gopi, and R. Maione. “The societal impact of commercial drones”. In: *Technology in Society* 45 (2016), pp. 83–90 (cit. on pp. 212, 213).

[522] T. S. Rappaport et al. “Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond”. In: *IEEE Access* 7 (2019), pp. 78729–78757 (cit. on p. 246).

[523] T. Rausch, T. Zichner, A. Schlattl, A. M. Stütz, V. Benes, and J. O. Korbel. “DELLY: structural variant discovery by integrated paired-end and split-read analysis”. In: *Bioinformatics* 28.18 (Sept. 2012), pp. i333–i339. URL: https://doi.org/10.1093/bioinformatics/bts378 (cit. on p. 55).

[524] S. Reddy, D. Estrin, M. Hansen, and M. Srivastava. “Examining micro-payments for participatory sensing data collections”. In: *Procs. of the ACM Int. Conference on Ubiquitous Computing 2010*. 2010 (cit. on p. 304).

[525] A. Rejeb, K. Rejeb, S. J. Simske, and H. Treiblmaier. “Drones for supply chain management and logistics: a review and research agenda”. In: *International Journal of Logistics Research and Applications* (2021), pp. 1–24 (cit. on p. 213).

[526] C. W. Reynolds. “Flocks, herds and schools: A distributed behavioral model”. In: *Procs. of the Conference on Computer Graphics and Interactive Techniques 1987*. 1987, pp. 25–34 (cit. on p. 214).

[527] M. T. Ribeiro, S. Singh, and C. Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2016*. KDD ’16. New York, NY, USA: ACM, 2016, pp. 1135–1144. URL: http://doi.acm.org/10.1145/2939672.2939778 (cit. on p. 17).

[528] M. T. Ribeiro, S. Singh, and C. Guestrin. “Anchors: High-precision model-agnostic explanations”. In: *Procs. of the AAAI Conference on Artificial Intelligence 2018*. Vol. 32. 1. 2018 (cit. on p. 17).

[529] M. T. Ribeiro, S. Singh, and C. Guestrin. “Model-agnostic Interpretability of Machine Learning”. In: *arXiv: Statistics* (2016). DOI: arXiv:1606.05386 (cit. on p. 17).

[530] J. Richter, H. Kotthaus, B. Bischl, P. Marwedel, J. Rahnenführer, and M. Lang. “Faster Model-Based Optimization through Resource-Aware Scheduling Strategies”. In: *Procs. of the Int. Conference on Learning and Intelligent Optimization 2010*. Vol. 10079. Lecture Notes in Computer Science (LNCS). Springer Int. Publishing, 2016, pp. 267–273. URL: http://link.springer.com/chapter/10.1007/978-3-319-50349-3_22 (cit. on p. 77). **SFB876-A3**

[531] J. Richter, K. Madjar, and J. Rahnenführer. “Model-based optimization of subgroup weights for survival analysis”. In: *Bioinformatics* 35.14 (July 2019), pp. i484–i491. URL: https://doi.org/10.1093/bioinformatics/btz361 (cit. on pp. 72, 78, 81, 83). **SFB876-A3**

[532] M. S. Ridgely and M. D. Greenberg. “Too many alerts, too much liability: sorting through the malpractice implications of drug-drug interaction clinical decision support”. In: *Saint Louis University Journal of Health Law & Policy* 5 (2011), p. 257 (cit. on p. 19).

[533] N. Rieke et al. “The future of digital health with federated learning”. In: *NPJ digital medicine* 3.1 (2020), pp. 1–7 (cit. on p. 14).

[534] K. Riesen, M. Ferrer, and H. Bunke. “Approximate Graph Edit Distance in Quadratic Time”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2015). Epub ahead of print (cit. on pp. 88, 102).

[535] K. Riesen and H. Bunke. “Approximate graph edit distance computation by means of bipartite graph matching”. In: *Image and Vision Computing* 27.7 (2009), pp. 950–959 (cit. on pp. 88, 94).

[536] A. P. Rifai, H. Aoyama, N. H. Tho, S. Z. M. Dawal, and N. A. Masruroh. “Evaluation of turned and milled surfaces roughness using convolutional neural network”. In: *Measurement* 161 (2020), p. 107860 (cit. on p. 158).

[537] E. Rivers et al. “Early goal-directed therapy in the treatment of severe sepsis and septic shock”. In: *New England Journal of Medicine* 345.19 (2001), pp. 1368–1377 (cit. on p. 5).

[538] A. Rizk-Jackson et al. “Evaluating imaging biomarkers for neurodegeneration in pre-symptomatic Huntington’s disease using machine learning techniques”. In: *Neuroimage* 56.2 (2011), pp. 788–796 (cit. on p. 10).

[539] S. J. Rizvi and J. R. Haritsa. “Maintaining data privacy in association rule mining”. In: *Procs. of the Int. Conference on Very Large Data Bases 2002*. Elsevier. 2002, pp. 682–693 (cit. on p. 15).

[540] Robert B Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. *STL: A Seasonal-Trend decomposition Procedure Based on Loess*. 1990 (cit. on p. 324).

[541] E. Roberts, B. A. Bassett, and M. Lochner. “Bayesian Anomaly Detection and Classification”. In: *arXiv: Statistics* (2019). DOI: arXiv:1902.08627 (cit. on p. 14).

[542] B. Røder, S. Nielsen, P. Magnussen, A. Engquist, and N. Frimodt-Møller. “Antibiotic usage in an intensive care unit in a Danish university hospital.” In: *The Journal of antimicrobial chemotherapy* 32.4 (1993), pp. 633–642 (cit. on p. 7).

[543] C. Römer, J. Hiry, C. Kittl, T. Liebig, and C. Rehtanz. “Charging control of electric vehicles using contextual bandits considering the electrical distribution grid”. In: *Procs. of the Int. Workshop on Knowledge Discovery from Mobility and Transportation Systems, co-located with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery 2018*. 2018, (accepted) (cit. on p. 240). **SFB876-B4**

[544] O. Ronneberger, P.Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Vol. 9351. LNCS. Springer, 2015, pp. 234–241 (cit. on p. 39).

[545] R. N. Rooks, E. M. Simonsick, L. M. Klesges, A. B. Newman, H. N. Ayonayon, and T. B. Harris. “Racial disparities in health care access and cardiovascular disease indicators in Black and White older adults in the Health ABC Study”. In: *Journal of aging and health* 20.6 (2008), pp. 599–614 (cit. on p. 18).

[546] A. Roth and T. Liebig. “Transforming PageRank into an Infinite-Depth Graph Neural Network”. In: *Procs. of the European Conference on Machine Learning and Knowledge Discovery in Databases 2022*. Springer. 2022, (accepted) (cit. on p. 240).

[547] A. Roth, K. Wüstefeld, and F. Weichert. “A Data-Centric Augmentation Approach for Disturbed Sensor Image Segmentation”. In: *Journal of Imaging* 7.10 (Oct. 2021). DOI: 10.3390/jimaging7100206. URL: https://www.mdpi.com/2313-433X/7/10/206 (cit. on pp. 38–40). **SFB876-B2, SFB876-B4**

[548] A. Roth. *Einführung und Umsetzung von Industrie 4.0: Grundlagen, Vorgehensmodell und Use Cases aus der Praxis*. Springer-Verlag, 2016 (cit. on pp. 103, 104).

[549] M. C. P. Rubio et al. “A Realistic Evaluation of Indoor Robot Position Tracking Systems: The IPIN 2016 Competition Experience”. In: *Measurement* 135 (Mar. 2019), pp. 151–162. URL: http://www.sciencedirect.com/science/article/pii/S0263224118310728 (cit. on p. 192).

[550] I. Ruczinski, C. Kooperberg, and M. LeBlanc. “Logic Regression”. In: *Journal of Computational and Graphical Statistics* 12.3 (2003), pp. 475–511 (cit. on p. 60).

[551] A. Ruepp et al. “CORUM: the comprehensive resource of mammalian protein complexes – 2009”. In: *Nucleic acids research* 38.suppl 1 (2010), pp. D497–D501 (cit. on pp. 87, 95).

[552] S. Rüping. “SVM Classifier Estimation From Group Probabilities”. In: *Procs. of the Int. Conference on Machine Learning 2010*. 2010, pp. 911–918 (cit. on pp. 140, 141, 151, 152).

[553] H. Ruppel, L. De Vaux, D. Cooper, S. Kunz, B. Duller, and M. Funk. “Testing physiologic monitor alarm customization software to reduce alarm rates and improve nurses’ experience of alarms in a medical intensive care unit”. In: *PloS one* 13.10 (2018), e0205901 (cit. on p. 5).

[554] O. Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115.3 (2015), pp. 211–252 (cit. on p. 8).

[555] J. Russell. *Frequently Asked Questions. The American Road and Transportation Builders Association (ARTBA)*. URL: https://www.artba.org/about/ (cit. on p. 297).

[556] A. Saadallah, L. Moreira-Matias, R. Sousa, J. Khiari, and J. Jenelius E.and Gama. “BRIGHT - Drift-Aware Demand Predictions for Taxi Networks”. In: *Procs. of the IEEE Int. Conference on Data Engineering 2018*. IEEE Transactions on Knowledge and Data Engineering, 2018 (cit. on p. 169). **SFB876-B3**

[557] A. Saadallah, O. Abdulaaty, J. Büscher, T. Panusch, K. Morik, and J. Deuse. “Early Quality Prediction using Deep Learning on Time Series Sensor Data”. In: *Procs. of the CIRP Conference on Manufacturing Systems 2022*. Elsevier, 2022 (cit. on p. 135). **SFB876-B3**

[558] A. Saadallah, J. Büscher, O. Abdulaaty, T. Panusch, J. Deuse, and K. Morik. “Explainable Predictive Quality Inspection using Deep Learning inElectronics Manufacturing”. In: *Procs. of the CIRP Conference on Manufacturing Systems 2022*. Elsevier, 2022 (cit. on p. 135). **SFB876-B3**

[559] A. Saadallah, A. Egorov, B.-T. Cao, S. Freitag, K. Morik, and G. Meschke. “Active Learning for Accurate Settlement Prediction Using Numerical Simulations in Mechanized Tunneling”. In: *CIRP Manufacturing Systems Conference 2019* (2019). URL: https://www.cirp-cms2019.org/ (cit. on pp. 160, 161). **SFB876-B3**

[560] A. Saadallah, F. Finkeldey, K. Morik, and P. Wiederkehr. “Stability prediction in milling processes using a simulation-based machine learning approach”. In: *Procs. of the CIRP Conference on Manufacturing Systems 2018*. Elsevier, 2018 (cit. on pp. 158–160, 162–165, 171). **SFB876-B3**

[561] A. Saadallah, M. Jakobs, and K. Morik. “Explainable Online Deep Neural Network Selection using Adaptive Saliency Maps for Time Series Forecasting”. In: *Procs. of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2021*. To appear. 2021 (cit. on p. 127). **SFB876-B3**

[562] A. Saadallah and M. Katharina. “Online Ensemble Aggregation using Deep Reinforcement Learning for Time Series Forecasting”. In: 2021 (cit. on p. 127). **SFB876-B3**

[563] A. Saadallah, N. Piatkowski, F. Finkeldey, P. Wiederkehr, and K. Morik. “Learning Ensembles in the Presence of Imbalanced Classes”. In: *ICPRAM: International Conference on Pattern Recognition Applications and Methods 2019* (2019). URL: http://www.icpram.org/Home.aspx (cit. on p. 125).

[564] A. Saadallah, F. Priebe, and M. Katharina. “A Drift-based Dynamic Ensemble Members Selection using Clustering for Time Series Forecasting”. In: *Procs. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2019*. Sept. 2019 (cit. on pp. 120, 127). **SFB876-B3**

[565] A. Saadallah, M. Tavakol, and M. Katharina. “An Actor-Critic Ensemble Aggregation Model for Time-Series Forecasting”. In: 2021 (cit. on p. 127). **SFB876-B3**

[566] A. Saadi, D. U. Himmelstein, S. Woolhandler, and N. I. Mejia. “Racial disparities in neurologic health care access and utilization in the United States”. In: *Neurology* 88.24 (2017), pp. 2268–2275 (cit. on p. 18).

[567] A. Sachan, D. Roy, and P. Arun. “An analysis of privacy preservation techniques in data mining”. In: *Advances in computing and information technology*. Springer, 2013, pp. 119–128 (cit. on p. 15).

[568] T. Sachweh, D. Boiar, and T. Liebig. “Differentially Private Learning from Label Proportions”. In: *Procs. of the ECML Workshop on Parallel, Distributed, and Federated Learning 2021*. 2021, accepted (cit. on p. 233). **SFB876-B4**

[569] A. Sadeghnejad Barkousaraie, O. Ogunmolu, S. Jiang, and D. Nguyen. “A fast deep learning approach for beam orientation optimization for prostate cancer treated with intensity-modulated radiation therapy”. In: *Medical physics* 47.3 (2020), pp. 880–897 (cit. on p. 6).

[570] A. Sadeghnejad-Barkousaraie, G. Bohara, S. Jiang, and D. Nguyen. “A reinforcement learning application of a guided Monte Carlo Tree Search algorithm for beam orientation selection in radiation therapy”. In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035013 (cit. on p. 8).

[571] R. Saket, A. Raghuveer, and B. Ravindran. “On Combining Bags to Better Learn from Label Proportions”. In: *Procs. of the Int. Conference on AI and Statistics 2022*. PMLR, 2022 (cit. on p. 139).

[572] J. Saleema, P. D. Shenoy, K. Venugopal, and L. Patnaik. “Cancer prognosis prediction model using data mining techniques”. In: *Data Mining and Knowledge Engineering* 6.1 (2014), pp. 21–29 (cit. on p. 15).

[573] C. Sánchez Claros and A. Tramontano. “Detecting mutually exclusive interactions in protein-protein interaction maps”. In: *PLoS One* 7.6 (2012), e38765 (cit. on p. 95).

[574] A. Sanfeliu and K. S. Fu. “A Distance measure between attributed relational graphs for pattern recognition.” In: *IEEE Transactions on Systems, Man, and Cybernetics* 13.3 (1983), pp. 353–362 (cit. on p. 87).

[575] R. Sardar, A. Sharma, and D. Gupta. “Machine learning assisted prediction of prognostic biomarkers associated with COVID-19, using clinical and proteomics data”. In: *Frontiers in genetics* 12 (2021) (cit. on p. 11).

[576] R. E. Schapir. “A brief introduction to boosting”. In: *Procs. of the Int. Joint Conference on Artificial Intelligence 1999*. 1999 (cit. on p. 125).

[577] R. B. M. Schasfoort. *Handbook of Surface Plasmon Resonance*. RSC Publishing, 2008 (cit. on p. 24).

[578] M. Schirmer, R. D’Amore, U. Z. Ijaz, N. Hall, and C. Quince. “Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data”. In: *BMC Bioinformatics* 17.1 (Mar. 2016), p. 125. URL: https://doi.org/10.1186/s12859-016-0976-y (cit. on p. 49).

[579] J. Schmitt, J. Boenig, T. Borggraefe, G. Beitinger, and J. Deuse. “Predictive model-based quality inspection in electronics manufacturing using Machine Learning and Edge Cloud Computing”. In: *Advanced Engineering Informatics (ADVEI)* (2020) (cit. on pp. 116, 128, 130). **SFB876-B3**

[580] J. Schmitt and J. Deuse. “Modellbasierte Prüfprozesse. Einsatz von Data-Mining-Verfahren zur industriellen Qualitätssicherung”. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF)* 114.4 (Apr. 2019), pp. 191–193 (cit. on pp. 128, 130, 131). **SFB876-B3**

[581] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. “Estimating the Support of a High-Dimensional Distribution”. In: *Neural Computation* 13.7 (July 2001), pp. 1443–1471 (cit. on p. 125).

[582] B. Schowe and K. Morik. “Fast-Ensembles of Minimum Redundancy Feature Selection”. In: *Ensembles in Machine Learning Applications*. Ed. by O. Okun, G. Valentini, and M. Re. Studies in Computational Intelligence. Springer, 2011, pp. 75–95 (cit. on p. 122).

[583] A. Schramm et al. “Mutational dynamics between primary and relapse neuroblastomas”. In: *Nature Genetics* 47.8 (Aug. 2015), pp. 872–877. URL: http://www.nature.com/ng/journal/vaop/ncurrent/full/ng.3349.html (cit. on pp. 44, 53, 54, 56). **SFB876-C1**

[584] G. Schuh, R. Anderl, J. Gausemeier, M. ten Hompel, and W. Wahlster. *Acatech Studie: Industrie 4.0 Maturity Index – Die Digitale Transformation von Unternehmen Gestalten*. 2017 (cit. on pp. 105, 106).

[585] C. Schüler, M. Patchou, B. Sliwa, and C. Wietfeld. “Robust machine learning-enabled routing for highly mobile vehicular networks with PARRoT in ns-3”. In: *Procs. of the Workshop on Ns-3*. Virtual, June 2021 (cit. on p. 276). **SFB876-A4, SFB876-B4**

[586] C. Schüler, B. Sliwa, and C. Wietfeld. “Towards Machine Learning-Enabled Context Adaption for Reliable Aerial Mesh Routing”. In: *Procs. of the IEEE Vehicular Technology Conference 2021*. Virtual Event, Sept. 2021 (cit. on pp. 274, 277, 278, 282, 283). **SFB876-A4, SFB876-B4**

[587] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. “Proximal Policy Optimization Algorithms”. In: *arXiv: Computing Research Repository* (2017). DOI: arXiv:1707.06347 (cit. on p. 223).

[588] M. Schulte, J. Köster, S. Rahmann, and A. Schramm. “Cancer evolution, mutations, and clonal selection in relapse neuroblastoma”. In: *Cell Tissue Research* 372.2 (May 2018), pp. 263–268 (cit. on p. 44). **SFB876-C1**

[589] M. Schumacher, H. Binder, and T. Gerds. “Assessment of survival prediction models based on microarray data”. In: *Bioinformatics* 23.14 (2007), pp. 1768–1774 (cit. on p. 65).

[590] P. Schwab et al. “Real-time prediction of COVID-19 related mortality using electronic health records”. In: *Nature communications* 12.1 (2021), pp. 1–16 (cit. on pp. 10, 11).

[591] N. Semiconductor. *nRF51822 Product Specification*. Tech. rep. Technical report, Nordic Semi, 2014 (cit. on p. 216).

[592] Semtech. *SX1272/73 -860 MHz to 1020 MHz Low Power Long Range Transceiver*. Tech. rep. Rev. 3.1. 2017, p. 129 (cit. on p. 318).

[593] B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012 (cit. on pp. 161, 163).

[594] A. Severyn and A. Moschitti. “Automatic feature engineering for answer selection and extraction”. In: *Procs. of the Conference on Empirical Methods in Natural Language Processing 2013*. 2013, pp. 458–467 (cit. on p. 122).

[595] A. Shahrjooihaghighi, H. Frigui, X. Zhang, X. Wei, B. Shi, and A. Trabelsi. “An ensemble feature selection method for biomarker discovery”. In: *Procs. of the IEEE Int. Symposium on Signal Processing and Information Technology 2017*. IEEE. 2017, pp. 416–421 (cit. on p. 9).

[596] C. Shang, A. Palmer, J. Sun, K.-S. Chen, J. Lu, and J. Bi. “VIGAN: Missing view imputation with generative adversarial networks”. In: *Procs. of the IEEE Int. Conference on Big Data 2017*. IEEE. 2017, pp. 766–775 (cit. on p. 13).

[597] X. Shao et al. “Conditional sum-product networks: Imposing structure on deep probabilistic architectures”. In: *Procs. of the Int. Conference on Probabilistic Graphical Models 2020*. PMLR. 2020, pp. 401–412 (cit. on pp. 231, 240). **SFB876-B4**

[598] X. Shao et al. “Conditional Sum-Product Networks: Modular Probabilistic Circuits via Gate Functions”. In: *International Journal of Approximate Reasoning* (2022), pp. 298–313 (cit. on p. 231). **SFB876-B4**

[599] S. Sharma, H. Mohammadmoradi, M. Heydariaan, and O. Gnawali. “Device-free activity recognition using ultra-wideband radios”. In: *Procs. of the Int. Conference on Computing, Networking and Communications 2019*. 2019, pp. 1029–1033 (cit. on p. 246).

[600] Z. Shelby et al. “The Constrained Application Protocol”. In: *Internet Engineering Task Force (IETF)* (2014) (cit. on p. 266).

[601] Z. Shen. “Cancer biomarkers and targeted therapies”. In: *Cell & bioscience* 3.1 (2013), pp. 1–2 (cit. on p. 9).

[602] N. Shervashidze, P. Schweitzer, E. van Leeuwen, K. Mehlhorn, and K. Borgwardt. “Weisfeiler–Lehman Graph Kernels”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2539–2561 (cit. on pp. 89–91).

[603] S. Shiraishi and K. L. Moore. “Knowledge-based prediction of three-dimensional dose distributions for external beam radiotherapy”. In: *Medical physics* 43.1 (2016), pp. 378–387 (cit. on p. 8).

[604] W. Shockley and W. T. Read. “Statistics of the Recombinations of Holes and Electrons”. In: 87.5 (1952), pp. 835–842. URL: https://link.aps.org/doi/10.1103/PhysRev.87.835 (cit. on p. 197).

[605] C. Shorten and T. M. Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of Big Data* 6.1 (2019), pp. 1–48 (cit. on p. 177).

[606] V. Shpacovitch and R. Hergenröder. “Optical and surface plasmonic approaches to characterize extracellular vesicles. A review”. In: *Analytica Chimica Acta* 1005 (2017), pp. 1–15. URL: https://doi.org/10.1016/j.aca.2017.11.066 (cit. on p. 26). **SFB876-B2**

[607] V. Shpacovitch and R. Hergenröder. “Optical and surface plasmonic approaches to characterize extracellular vesicles. A review”. In: *Analytica Chimica Acta* 1005 (2018), pp. 1–15. URL: https://www.sciencedirect.com/science/article/pii/S0003267017313594 (cit. on p. 21).

[608] V. Shpacovitch et al. “Application of Surface Plasmon Resonance Imaging Technique for the Detection of Single Spherical Biological Submicron-particles”. In: *Analytical Biochemistry: Methods in the Biological Sciences* (2015) (cit. on p. 25). **SFB876-B2**

[609] V. Shpacovitch et al. “Application of the PAMONO-sensor for Quantification of Microvesicles and Determination of Nano-particle Size Distribution”. In: *Sensors* 17.2 (2017), pp. 1–14. URL: http://dx.doi.org/10.3390/s17020244 (cit. on pp. 26, 37). **SFB876-B2, SFB876-C1**

[610] S. N. Shukla and B. M. Marlin. “Interpolation-Prediction Networks for Irregularly Sampled Time Series”. In: *arXiv: Computing Research Repository* (2019). DOI: arXiv:1909.07782 (cit. on p. 13).

[611] G. L. Sicherman, W. de Jonge, and R. P. van de Riet. “Answering queries without revealing secrets”. In: *ACM Transactions on Database Systems* 8.1 (1983), pp. 41–59 (cit. on p. 411).

[612] A. J. Siddiqui, A. Mammeri, and A. Boukerche. “Towards Efficient Vehicle Classification in Intelligent Transportation Systems”. In: *Procs. of the ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications 2015*. DIVANet ’15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 19–25. DOI: https://doi.org/10.1145/2815347.2815354 (cit. on p. 245).

[613] D. Siedhoff. “A Parameter-Optimizing Model-Based Approach to the Analysis of Low-SNR Image Sequences for Biological Virus Detection”. PhD thesis. Dortmund, Germany: TU Dortmund University, 2016. DOI: http://dx.doi.org/10.17877/DE290R-17272 (cit. on pp. 22, 29, 30, 34, 35). **SFB876-B2**

[614] D. Siedhoff, P. Libuschewski, F. Weichert, A. Zybin, P. Marwedel, and H. Müller. “Modellierung und Optimierung eines Biosensors zur Detektion viraler Strukturen”. In: *Bildverarbeitung für die Medizin 2014*. Springer Berlin Heidelberg, 2014, pp. 108–113. URL: http://dx.doi.org/10.1007/978-3-642-54111-7_24 (cit. on p. 27). **SFB876-B2**

[615] D. Siedhoff, F. Weichert, P. Libuschewski, and C. Timm. “Detection and Classification of Nano-Objects in Biosensor Data”. In: *Procs. of the Microscopic Image Analysis with Applications in Biology Workshop 2011*. 2011 (cit. on p. 30). **SFB876-B2**

[616] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal. “Cancer statistics, 2021.” In: *CA: a cancer journal for clinicians* 71.1 (2021), pp. 7–33 (cit. on p. 7).

[617] D. Siepmann. “Industrie 4.0 – Grundlagen und Gesamtzusammenhang”. In: *Einführung und Umsetzung von Industrie 4.0: Grundlagen, Vorgehensmodell und Use Cases aus der Praxis*. Ed. by A. Roth. Springer, 2016, pp. 17–82. URL: https://doi.org/10.1007/978-3-662-48505-7_2 (cit. on pp. 104, 105).

[618] J. H. Silber, P. S. Romano, A. K. Rosen, Y. Wang, O. Even-Shoshan, and K. G. Volpp. “Failure-to-rescue: comparing definitions to measure quality of care”. In: *Medical care* (2007), pp. 918–925 (cit. on p. 11).

[619] K. R. Simpson and A. Lyndon. “False alarms and overmonitoring: major factors in alarm fatigue among labor nurses”. In: *Journal of nursing care quality* 34.1 (2019), pp. 66–72 (cit. on p. 5).

[620] D. Sinwar and M. Kumar. “Anomaly detection using decision tree based classifiers”. In: *International Journal of modern Trends in Engineering and Research(IJMTER)* 3 (2016), pp. 1–4 (cit. on pp. 12, 14).

[621] B. Sliwa, R. Adam, and C. Wietfeld. “Acting selfish for the good of all: Contextual bandits for resource-efficient transmission of vehicular sensor data”. In: *Procs. of the ACM MobiHoc Workshop on Cooperative Data Dissemination in Future Vehicular Networks 2020*. Online, Oct. 2020 (cit. on p. 243). **SFB876-B4**

[622] B. Sliwa, R. Adam, and C. Wietfeld. “Client-Based Intelligence for Resource Efficient Vehicular Big Data Transfer in Future 6G Networks”. In: *IEEE Transactions on Vehicular Technology* (Feb. 2021). URL: https://arxiv.org/abs/2102.08624 (cit. on pp. 334, 338, 352). **SFB876-A4, SFB876-B4**

[623] B. Sliwa, D. Behnke, C. Ide, and C. Wietfeld. “B.A.T.Mobile: Leveraging mobility control knowledge for efficient routing in mobile robotic networks”. In: *Procs. of the IEEE GLOBECOM Workshop on Wireless Networking, Control and Positioning of Unmanned Autonomous Vehicles 2016*. Washington D.C., USA: IEEE, Dec. 2016. URL: http://dx.doi.org/10.1109/GLOCOMW.2016.7848845 (cit. on p. 273). **SFB876-B4**

[624] B. Sliwa, R. Falkenberg, T. Liebig, N. Piatkowski, and C. Wietfeld. “Boosting Vehicle-to-cloud Communication by Machine Learning-enabled Context Prediction”. In: *IEEE Transactions on Intelligent Transportation Systems* (2019) (cit. on p. 279). **SFB876-B4, SFB876-A4, SFB876-A1**

[625] B. Sliwa, R. Falkenberg, and C. Wietfeld. “Machine learning for resource-efficient data transfer in mobile crowdsensing”. In: *Machine Learning for Future Wireless Communications*. Ed. by F.-L. Luo. John Wiley & Sons Inc, Mar. 2020 (cit. on p. 334). **SFB876-A4, SFB876-B4**

[626] B. Sliwa, R. Falkenberg, and C. Wietfeld. “Towards cooperative data rate prediction for future mobile and vehicular 6G networks”. In: *Procs. of the 6G Wireless Summit 2020*. Levi, Finland: IEEE, Mar. 2020 (cit. on pp. 340, 352, 353). **SFB876-A4, SFB876-B4**

[627] B. Sliwa, S. Falten, and C. Wietfeld. “Performance evaluation and optimization of B.A.T.M.A.N. V routing for aerial and ground-based mobile ad-hoc networks”. In: *Procs. of the IEEE Vehicular Technology Conference 2019*. Kuala Lumpur, Malaysia, Apr. 2019. URL: http://arxiv.org/abs/1901.02298 (cit. on p. 273). **SFB876-B4**

[628] B. Sliwa, M. Haferkamp, M. Al-Askary, and C. Wietfeld. “A radio-fingerprinting-based vehicle classification system for intelligent traffic control in smart cities”. In: *Procs. of the IEEE Int. Systems Conference 2018*. Vancouver, Canada: IEEE, Apr. 2018 (cit. on p. 334). **SFB876-B4**

[629] B. Sliwa, T. Liebig, R. Falkenberg, J. Pillmann, and C. Wietfeld. “Efficient machine-type communication using multi-metric context-awareness for cars used as mobile sensors in upcoming 5G networks”. In: *Procs. of the IEEE Vehicular Technology Conference 2018*. Best Student Paper Award. IEEE. 2018 (cit. on p. 334). **SFB876-B4, SFB876-A4**

[630] B. Sliwa, T. Liebig, T. Vranken, M. Schreckenberg, and C. Wietfeld. "System-of-systems modeling, analysis and optimization of hybrid vehicular traffic". In: *Procs. of the IEEE Int. Systems Conference 2019*. Orlando, Florida, USA: IEEE, Apr. 2019 (cit. on p. 332). **SFB876-B4**

[631] B. Sliwa, M. Patchou, K. Heimann, and C. Wietfeld. "Simulating hybrid aerial- and ground-based vehicular networks with ns-3 and LIMoSim". In: *Procs. of the Workshop on Ns-3 2020*. Gaithersburg, Maryland, USA, June 2020 (cit. on p. 385). **SFB876-B4**

[632] B. Sliwa, M. Patchou, and C. Wietfeld. "Lightweight simulation of hybrid aerial- and ground-based vehicular communication networks". In: *Procs. of the IEEE Vehicular Technology Conference 2019*. Honolulu, Hawaii, USA, Sept. 2019 (cit. on p. 334). **SFB876-B4**

[633] B. Sliwa, N. Piatkowski, and C. Wietfeld. "LIMITS: Lightweight machine learning for IoT systems with resource limitations". In: *Procs. of the IEEE Int. Conference on Communications 2020*. Best paper award. IEEE. Dublin, Ireland, June 2020, pp. 1–7. URL: https://arxiv.org/abs/2001.10189 (cit. on pp. 277, 335). **SFB876-B4**

[634] B. Sliwa, N. Piatkowski, and C. Wietfeld. "The channel as a traffic sensor: Vehicle detection and classification based on radio fingerprinting". In: *IEEE Internet of Things Journal* (Mar. 2020) (cit. on pp. 243, 244, 247–249, 253, 255–258). **SFB876-B4**

[635] B. Sliwa, C. Schüler, M. Patchou, and C. Wietfeld. "PARRoT: Predictive ad-hoc routing fueled by reinforcement learning and trajectory knowledge". In: *Procs. of the IEEE Vehicular Technology Conference 2021*. Helsinki, Finland, Apr. 2021 (cit. on pp. 273, 279). **SFB876-A4, SFB876-B4**

[636] B. Sliwa and C. Wietfeld. "A Reinforcement Learning Approach for Efficient Opportunistic Vehicle-to-Cloud Data Transfer". In: *Procs. of the IEEE Wireless Communications and Networking Conference 2020*. Seoul, South Korea, Apr. 2020. DOI: https://doi.org/10.48550/arXiv.2001.05321 (cit. on p. 334). **SFB876-B4**

[637] B. Sliwa and C. Wietfeld. "LIMoSim: A framework for lightweight simulation of vehicular mobility in intelligent transportation systems". In: *Recent Advances in Network Simulation - The OMNeT++ Environment and its Ecosystem*. Ed. by A. Virdis and M. Kirsche. Springer Int. Publishing, May 2019. Chap. LIMoSim: A framework for lightweight simulation of vehicular mobility in intelligent transportation systems. URL: https://www.springer.com/de/book/9783030128418%5C#aboutBook (cit. on p. 338). **SFB876-B4**

[638] B. Sliwa and C. Wietfeld. "Towards data-driven simulation of end-to-end network performance indicators". In: *Procs. of the IEEE Vehicular Technology Conference 2019*. Honolulu, Hawaii, USA, Sept. 2019 (cit. on pp. 333, 335). **SFB876-B4**

[639] I. Smal, M. Loog, W. Niessen, and E. Meijering. "Quantitative Comparison of Spot Detection Methods in Fluorescence Microscopy". In: *IEEE Transactions on Medical Imaging* 29.2 (2010), pp. 282–301 (cit. on p. 27).

[640] R. Sonabend, F. J. Király, A. Bender, B. Bischl, and M. Lang. "mlr3proba: An R Package for Machine Learning in Survival Analysis". In: *Bioinformatics* (Feb. 2021) (cit. on p. 83). **SFB876-A3**

[641] S. Song, C. Li, and X. Zhang. "Turn Waste into Wealth: On Simultaneous Clustering and Cleaning over Dirty Data". In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2015*. Ed. by L. Cao, C. Zhang, T. Joachims, G. I. Webb, D. D. Margineantu, and G. Williams. ACM, 2015, pp. 1115–1124. DOI: http://doi.acm.org/10.1145/2783258.2783317 (cit. on p. 12).

[642] F. B. Sorbelli, F. Corò, S. K. Das, and C. M. Pinotti. "Energy-constrained delivery of goods with drones under varying wind conditions". In: *IEEE Transactions on Intelligent Transportation Systems* 22.9 (2020), pp. 6048–6060 (cit. on p. 213).

[643] S. Srihari, C. H. Yong, and L. Wong. *Computational Prediction of Protein Complexes from Protein Interaction Networks*. New York, NY, USA: Association for Computing Machinery and Morgan & Claypool, 2017 (cit. on p. 86).

[644] B. K. St"ocker, T. Sch"afer, P. Mutzel, J. K"oster, N. M. Kriege, and S. Rahmann. "Protein Complex Similarity Based on Weisfeiler-Lehman Labeling". In: *Procs. of the Int. Conference on Similarity Search and Applications 2021*. Ed. by G. Amato, C. Gennaro, V. Oria, and M. Radovanovic. Cham: Springer International Publishing, 2019, pp. 308–322 (cit. on p. 86).

[645] A. Stanford-Clark et al. "MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2". In: *OASIS standard* (2013) (cit. on p. 266).

[646] M. R. Stavseth, T. Clausen, and J. Røislien. "How handling missing data may impact conclusions: A comparison of six different imputation methods for categorical questionnaire data". In: *SAGE open medicine* 7 (2019), p. 2050312118822912 (cit. on p. 13).

[647] D. J. Stekhoven and P. Bühlmann. "MissForest – non-parametric missing value imputation for mixed-type data". In: *Bioinformatics* 28.1 (2012), pp. 112–118 (cit. on p. 13).

[648] STMicroelectronics. *DM00037051 - STM32F405xx STM32F407xx Datasheet*. STMicroelectronics. 2020 (cit. on p. 216).

[649] B. K. Stöcker, J. Köster, E. Zamir, and S. Rahmann. "Modeling and simulating networks of interdependent protein interactions". In: *Integrative Biology* 10 (2018), pp. 290–305 (cit. on pp. 95, 102). **SFB876-C1**

[650] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley. "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery". In: *Nature Communications* 9.1 (Feb. 2018) (cit. on p. 212).

[651] M. Stolpe and K. Morik. "Learning from Label Proportions by Optimizing Cluster Model Selection". In: *Procs. of the Machine Learning and Knowledge Discovery in Databases European Conference 2011*. Ed. by D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis. Springer, 2011, pp. 349–364 (cit. on pp. 139, 152, 154–156, 231). **SFB876-B3**

[652] M. Stolpe. "Distributed Analysis of Vertically Partitioned Sensor Measurements under Communication Constraints". PhD thesis. Dortmund: TU Dortmund University, 2017. URL: http://dx.doi.org/10.17877/DE290R-17839 (cit. on pp. 139, 156). **SFB876-B3**

[653] M. Stolpe. "The Internet of Things: Opportunities and Challenges for Distributed Data Analysis". In: *SIGKDD Explorations* 18.1 (June 2016), pp. 15–34 (cit. on p. 137). **SFB876-B3**

[654] M. Stolpe, H. Blom, and K. Morik. "Sustainable Industrial Processes by Embedded Real-Time Quality Prediction". In: *Computational Sustainability*. Ed. by K. Kersting, J. Lässig, and K. Morik. Springer, 2016, pp. 201–243. URL: http://link.springer.com/chapter/10.1007/978-3-319-31858-5_10 (cit. on pp. 114, 116–122, 124, 132–134, 156). **SFB876-A1, SFB876-B3**

[655] M. Stolpe, T. Liebig, and K. Morik. "Communication-efficient learning of traffic flow in a network of wireless presence sensors". In: *Procs. of the Workshop on Parallel and Distributed Computing for Knowledge Discovery in Data Bases 2015*. CEUR Workshop Proceedings. CEUR-WS, 2015, (to appear) (cit. on pp. 150, 156, 231). **SFB876-B3**

[656] J. Stoyanovich, B. Howe, and H. Jagadish. "Responsible data management". In: *Procs. of the VLDB Endowment* 13.12 (2020), pp. 3474–3488 (cit. on p. 13).

[657] C. Sturm and W. Wiesbeck. "Waveform Design and Signal Processing Aspects for Fusion of Wireless Communications and Radar Sensing". In: *Procs. of the IEEE* 99.7 (July 2011), pp. 1236–1259 (cit. on p. 387).

[658] C. Sudlow et al. "UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age". In: *PLoS medicine* 12.3 (2015), e1001779 (cit. on p. 4).

[659] S. J. Sujit, I. Coronado, A. Kamali, P. A. Narayana, and R. E. Gabr. “Automated image quality evaluation of structural brain MRI using an ensemble of deep learning networks”. In: *Journal of Magnetic Resonance Imaging* 50.4 (2019), pp. 1260–1267 (cit. on p. 6).

[660] T. Sun, D. Sheldon, and B. O’Connor. “A Probabilistic Approach for Learning with Label Proportions Applied to the US Presidential Election”. In: *Procs. of the IEEE Int. Conference on Data Mining 2017*. 2017, pp. 445–454 (cit. on p. 139).

[661] M. Sundararajan and A. Najmi. “The many Shapley values for model explanation”. In: *Procs. of the Int. Conference on Machine Learning 2020*. PMLR. 2020, pp. 9269–9278 (cit. on p. 17).

[662] T. Surmann, D. Biermann, and G. Kehl. “Oscillator model of machine tools for the simulation of self excited vibrations in machining processes”. In: *Procs. of the Int. Conference on Process Machine Interactions 2008*. 2008, pp. 23–29 (cit. on pp. 160, 173).

[663] B. Suthar, H. Patel, and A. Goswami. “A survey: classification of imputation methods in data mining”. In: *International Journal of Emerging Technology and Advanced Engineering* 2.1 (2012), pp. 309–12 (cit. on p. 13).

[664] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cit. on p. 222).

[665] S. Suzuki and K. Abe. “Topological structural analysis of digitized binary images by border following”. In: *Computer vision, graphics, and image processing* 30.1 (1985), pp. 32–46 (cit. on p. 176).

[666] A. Swaminathan and T. Joachims. “Batch learning from logged bandit feedback through counterfactual risk minimization”. In: *Journal of Machine Learning Research* 16 (2015), pp. 1731–1755 (cit. on pp. 236–238).

[667] A. L. Swan, A. Mobasheri, D. Allaway, S. Liddell, and J. Bacardit. “Application of machine learning to proteomics data: classification and biomarker identification in postgenomics biology”. In: *Omics: a journal of integrative biology* 17.12 (2013), pp. 595–610 (cit. on p. 9).

[668] D. Talia and P. Trunfio. *Service-oriented Distributed Knowledge Discovery*. CRC Press, 2013 (cit. on p. 116).

[669] R. M. Talukder et al. “Modifications of the PAMONO-Sensor Help to Size and Quantify Low Number of Individual Biological and Non-Biological Nano-Particles”. In: *Engineering Proceedings* 6.1 (2021). URL: https://www.mdpi.com/2673-4591/6/1/26 (cit. on p. 25).

[670] F. Tang and H. Ishwaran. “Random forest missing data algorithms”. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 10.6 (2017), pp. 363–377 (cit. on p. 13).

[671] X. Tang, Z. Wang, Z. Xu, Y. Ma, H. Zhu, and J. Ye. “A Deep Value-network Based Approach for Multi-Driver Order Dispatching”. In: *Procs. of the Int. Conference on Knowledge Discovery and Data Mining 2019*. 2019 (cit. on p. 274).

[672] X. Tang et al. “On the origin and continuing evolution of SARS-CoV-2”. In: *National Science Review* 7.6 (2020), pp. 1012–1023 (cit. on p. 11).

[673] M. A. Tanner and W. H. Wong. “The calculation of posterior distributions by data augmentation”. In: *Journal of the American Statistical Association* 82.398 (1987), pp. 528–550 (cit. on p. 64).

[674] M. Al-Tarawneh, Y. Huang, P. Lu, and D. Tolliver. “Vehicle Classification System Using In-Pavement Fiber Bragg Grating Sensors”. In: *IEEE Sensors Journal* 18.7 (2018), pp. 2807–2815 (cit. on p. 245).

[675] D. M. J. Tax and R. P. W. Duin. “Support Vector Data Description”. In: *Machine Learning* 54.1 (Jan. 2004), pp. 45–66 (cit. on p. 125).

[676] H. Tersing et al. “Simulation of manufacturing chain of a titanium aerospace component with experimental validation”. In: *Finite Elements in Analysis and Design* 51.Supplement C (2012), pp. 10–21 (cit. on p. 158).

[677] R. Tibshirani. “The Lasso Method for Variable Selection in the Cox Model”. In: *Statistics in Medicine* 16.4 (1997), pp. 385–395 (cit. on p. 80).

[678] J. Tiemann and Y. Elmasry. *ATLAS localization framework ROS package repository*. Sept. 2018 (cit. on pp. 183, 185).

[679] J. Tiemann, Y. Elmasry, and L. Koring. *Dataset and Interactive Demo: ATLAS FaST multi-user tracking in heterogeneous logistic applications*. Sept. 2018 (cit. on pp. 183, 185).

[680] J. Tiemann, L. Koring, P. Gorczak, and C. Wietfeld. “Improving the Robustness of Control-Grade Ultra-Wideband Localization”. In: *Procs. of the IFAC Conference on Embedded Systems, Computer Intelligence and Telematics 2018*. Faro, Portugal, June 2018 (cit. on pp. 183, 187, 191).

[681] J. Tiemann, A. Ramsey, and C. Wietfeld. “Enhanced UAV Indoor Navigation through SLAM-Augmented UWB Localization”. In: *Procs. of the IEEE Int. Conference on Communications Workshops: The IEEE Int. Workshop on Communication, Computing, and Networking in Cyber Physical Systems 2018*. Kansas City, USA, May 2018 (cit. on p. 183).

[682] J. Tiemann, F. Schweikowski, and C. Wietfeld. “Design of an UWB Indoor-Positioning System for UAV Navigation in GNSS-Denied Environments”. In: *Procs. of the Int. Conference on Indoor Positioning and Indoor Navigation 2015*. Banff, Canada, Oct. 2015 (cit. on p. 182). **SFB876-A4**

[683] J. Tiemann. “Scalability, Reliability and Energy Efficiency in Ultra-Wideband Indoor Localizations”. PhD thesis. TU Dortmund University, Oct. 2016 (cit. on pp. 183, 188, 191).

[684] J. Tiemann, F. Eckermann, and C. Wietfeld. “ATLAS - An Open-Source TDOA-based Ultra-Wideband Localization System”. In: *Procs. of the Int. Conference on Indoor Positioning and Indoor Navigation 2016*. Alcala de Henares, Madrid, Spain, Oct. 2016 (cit. on pp. 182, 185, 190). **SFB876-A4**

[685] J. Tiemann, F. Eckermann, and C. Wietfeld. “Multi-User Interference and Wireless Clock Syncronization in TDOA-based UWB Localization”. In: *Procs. of the Int. Conference on Indoor Positioning and Indoor Navigation 2016*. Accepted for presentation. Alcala de Henares, Madrid, Spain, Oct. 2016 (cit. on pp. 182, 184). **SFB876-A4**

[686] J. Tiemann, J. Pillmann, and C. Wietfeld. “Ultra-Wideband Antenna-Induced Error Prediction using Deep Learning on Channel Response Data”. In: *Procs. of the IEEE Vehicular Technology Conference 2017*. Sydney, Australia, June 2017 (cit. on p. 188). **SFB876-A4**

[687] J. Tiemann and C. Wietfeld. “Scalability, Real-Time Capabilities and Energy Efficiency in Ultra-Wideband Localization”. In: *IEEE Transactions on Industrial Informatics* 15.8 (2019). forthcoming. URL: https://ieeexplore.ieee.org/document/8618347 (cit. on pp. 182, 183, 185, 188, 250). **SFB876-A4**

[688] J. Tiemann and C. Wietfeld. “Scalable and Precise Multi-UAV Indoor Navigation using TDOA-based UWB Localization”. In: *Procs. of the Int. Conference on Indoor Positioning and Indoor Navigation 2017*. Sept. 2017, pp. 1–7 (cit. on p. 182). **SFB876-A4**

[689] W. R. Tobler. “A computer movie simulating urban growth in the Detroit region”. In: *Economic geography* 46.sup1 (1970), pp. 234–240 (cit. on p. 227).

[690] M.-Q. Tran, M.-K. Liu, and Q.-V. Tran. “Milling chatter detection using scalogram and deep convolutional neural network”. In: *The International Journal of Advanced Manufacturing Technology* 107.3 (2020), pp. 1505–1516 (cit. on p. 158).

[691] T. Treppmann, K. Ickstadt, and M. Zucknick. “Integration of multiple genomic data sources in a Bayesian Cox model for variable selection and prediction”. In: *Computational and Mathematical Methods in Medicine* Vol. 2017 (2017), pp. 1–19. URL: https://www.hindawi.com/journals/cmmm/2017/7340565/ (cit. on pp. 62, 65). **SFB876-C4**

[692] O. Troyanskaya et al. “Missing value estimation methods for DNA microarrays”. In: *Bioinformatics* 17.6 (2001), pp. 520–525 (cit. on p. 13).

[693] A. J. Tudos and R. B. M. Schasfoort. “Introduction to surface plasmon resonance.” In: *Handbook of Surface Plasmon Resonance.* Ed. by A. J. Tudos and R. B. M. Schasfoort. RSC Publishing, 2008, pp. 1–13 (cit. on p. 24).

[694] N. Ueda and R. Nakano. “Generalization error of ensemble estimators”. In: *Procs. of the Int. Conference on Neural Networks 1996*. Vol. 1. IEEE. 1996, pp. 90–95 (cit. on p. 169).

[695] J. Um, I. Kim, and S. Park. “Implementation of Platform for Measurement and Analysis on LTE Traffic and Radio Resource Utilization”. In: *Procs. of the IEEE Int. Conference on Consumer Electronics 2019*. Jan. 2019, pp. 1–2 (cit. on p. 345).

[696] Z. Usfoor, K. Kaufmann, A. S. H. Rakib, R. Hergenröder, and V. Shpacovitch. “Features of Sizing and Enumeration of Silica and Polystyrene Nanoparticles by Nanoparticle Tracking Analysis (NTA)”. In: *Sensors* 20.22 (Nov. 2020), p. 6611. DOI: doi:10.3390/s20226611 (cit. on p. 26). **SFB876-B2**

[697] A. Ustundag and E. Cevikcan. *Industry 4.0: Managing the Digital Transformation*. Springer, 2017 (cit. on p. 103).

[698] T. Vaa, P. Melén, D. Andersson, and B. B. Nielsen. “NorSIKT – Nordic System for Intelligent Classification of Traffic”. In: *Procedia - Social and Behavioral Sciences* 48 (2012). Transport Research Arena 2012, pp. 1702–1712 (cit. on p. 244).

[699] S. Vaishnav, A. Agarwal, and K. Desai. “Machine learning-based instantaneous cutting force model for end milling operation”. In: *Journal of Intelligent Manufacturing* (2019), pp. 1–14 (cit. on p. 158).

[700] V. N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. 2nd. New York: Springer, 1999 (cit. on pp. 139, 151, 152).

[701] A. Varga and R. Hornig. “An overview of the OMNeT++ simulation environment”. In: *Procs. of the Int. Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops 2008*. Simutools ’08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, 60:1–60:10 (cit. on p. 273).

[702] A. Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30: Procs. of the 2017 Conference*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https : // proceedings . neurips . cc / paper / 2017 / file / 3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (cit. on pp. 11, 13).

[703] R. G. Vaughan. *Radiance Calculator program*. 2009 (cit. on p. 201).

[704] W. Viriyasitavat, M. Boban, H. Tsai, and A. Vasilakos. “Vehicular Communications: Survey and Challenges of Channel and Propagation Models”. In: *IEEE Vehicular Technology Magazine* 10.2 (2015), pp. 55–66 (cit. on p. 274).

[705] S. Vishwanathan, N. Schraudolph, R. Kondor, and K. Borgwardt. “Graph Kernels”. In: *Journal of Machine Learning Research* 11 (2010), pp. 1201–1242 (cit. on p. 102).

[706] P. Voigt and A. Von dem Bussche. “The EU General Data Protection Regulation (GDPR)”. In: *A Practical Guide, 1st Ed., Cham: Springer Int. Publishing* 10.3152676 (2017), pp. 10–5555 (cit. on p. 14).

[707] M. Volkovs, F. Chiang, J. Szlichta, and R. J. Miller. “Continuous data cleaning”. In: *Procs. of the IEEE Int. Conference on Data Engineering 2014*. IEEE. 2014, pp. 244–255 (cit. on p. 12).

[708] T. Vranken, B. Sliwa, C. Wietfeld, and M. Schreckenberg. “Adapting a cellular automata model to describe heterogeneous traffic with human-driven, automated, and communicating automated vehicles”. In: *Physica A: Statistical Mechanics and its Applications* (Feb. 2021) (cit. on pp. 293, 295). **SFB876-B4**

[709] T. Vranken, B. Sliwa, C. Wietfeld, and M. Schreckenberg. “Performance comparison of dynamic vehicle routing methods for minimizing the global dwell time in upcoming smart

cities". In: *Procs. of the IEEE Vehicular Technology Conference 2018*. Chicago, USA, Aug. 2018 (cit. on pp. 290, 291). **SFB876-B4**

[710] T. P. E. Vranken and M. Schreckenberg. "Cellular Automata Intersection Model". In: *Collective Dynamics* 5 (2020) (cit. on pp. 290, 291). **SFB876-B4**

[711] S. Wachter, B. Mittelstadt, and C. Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR". In: *Harvard Journal of Law & Technology* 31 (2017), p. 841 (cit. on p. 17).

[712] G. Wang, B. Wang, X. Yang, and G. Yu. "Efficiently Indexing Large Sparse Graphs for Similarity Search". In: *IEEE Transactions on Knowledge and Data Engineering* 24.3 (2012), pp. 440–451 (cit. on p. 93).

[713] M. Wang, Q. Zhang, S. Lam, J. Cai, and R. Yang. "A review on application of deep learning algorithms in external beam radiotherapy automated treatment planning". In: *Frontiers in oncology* 10 (2020), p. 2177 (cit. on pp. 5, 8).

[714] R. Wang, F. M. Harper, and H. Zhu. "Factors Influencing Perceived Fairness in Algorithmic Decision-Making: Algorithm Outcomes, Development Procedures, and Individual Differences". In: *Procs. of the Conference on Human Factors in Computing Systems 2020*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–14 (cit. on p. 332).

[715] S. Wang and X. Yao. "Diversity analysis on imbalanced data sets by using ensemble models". In: *2009 IEEE symposium on computational intelligence and data mining* (2009) (cit. on p. 125).

[716] S.-C. Wang. "Artificial neural network". In: *Interdisciplinary computing in java programming*. Springer, 2003, pp. 81–100 (cit. on p. 8).

[717] W. Wang, G.-S. Hong, Y. Wong, and K. Zhu. "Sensor fusion for online tool condition monitoring in milling". In: *International Journal of Production Research* 45.21 (2007), pp. 5095–5116 (cit. on p. 158).

[718] X. Wang, H. Zhao, and J. Zhu. "GRPC: A communication cooperation mechanism in distributed systems". In: *ACM SIGOPS Operating Systems Review* 27.3 (1993), pp. 75–86 (cit. on p. 223).

[719] S. Warnat-Herresthal et al. "Swarm Learning for decentralized and confidential clinical machine learning". In: *Nature* 594.7862 (2021), pp. 265–270 (cit. on pp. 14, 15).

[720] S. Watkins et al. "Ten questions concerning the use of drones in urban environments". In: *Building and Environment* 167 (2020), p. 106458 (cit. on p. 213).

[721] L. Wawrla, O. Maghazei, and T. Netland. "Applications of drones in warehouse operations". In: *Whitepaper. ETH Zurich, D-MTEC* (2019) (cit. on p. 212).

[722] C. Weis et al. "Direct Antimicrobial Resistance Prediction from clinical MALDI-TOF mass spectra using Machine Learning". In: *bioRxiv* (2021). URL: https://www.biorxiv.org/content/early/2021/07/12/2020.07.30.228411 (cit. on pp. 5–7).

[723] C. V. Weis, C. R. Jutzeler, and K. Borgwardt. "Machine learning for microbial identification and antimicrobial susceptibility testing on MALDI-TOF mass spectra: a systematic review". In: *Clinical Microbiology and Infection* (2020) (cit. on p. 7).

[724] B. Weisfeiler and A. A. Lehman. "A Reduction of a Graph to a Canonical Form and an Algebra Arising During this Reduction". In: *Nauchno-Technicheskaya Informatsia* 2.9 (1968) (cit. on pp. 88, 89).

[725] D. Westreich. "Berkson's bias, selection bias, and missing data". In: *Epidemiology (Cambridge, Mass.)* 23.1 (2012), p. 159 (cit. on p. 13).

[726] V. Weyer and H. Binder. "A Weighting Approach for Judging the Effect of Patient Strata on High-Dimensional Risk Prediction Signatures". In: *BMC bioinformatics* 16 (Sept. 2015), p. 294 (cit. on pp. 78, 79).

[727] P. Wiederkehr, F. Finkeldey, and T. Merhofe. “Augmented semantic segmentation for the digitization of grinding tools based on deep learning”. In: *CIRP Annals* 70.1 (2021), pp. 297–300 (cit. on pp. 172, 176–178). **SFB876-B3**

[728] P. Wiederkehr and T. Siebrecht. “Virtual Machining: Capabilities and Challenges of Process Simulations in the Aerospace Industry”. In: *Procedia Manufacturing* 6 (2016), pp. 80–87 (cit. on pp. 157–160).

[729] C. Williams, E. V. Bonilla, and K. M. Chai. “Multi-task Gaussian process prediction”. In: *Advances in Neural Information Processing Systems 20: Procs. of the 2007 Conference*. 2007, pp. 153–160 (cit. on p. 13).

[730] M. J. Williams, B. Werner, C. P. Barnes, T. A. Graham, and A. Sottoriva. “Identification of neutral tumor evolution across cancer types”. In: *Nature Genetics* 48.3 (Mar. 2016), pp. 238–244. URL: https://www.nature.com/articles/ng.3489 (cit. on p. 51).

[731] I. H. Witten, F. Eibe, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. Data Management Systems. Burlington, MA: Elsevier, Inc., 2011 (cit. on pp. 137, 148).

[732] M. Won, S. Sahu, and K. Park. “DeepWiTraffic: Low Cost WiFi-Based Traffic Monitoring System Using Deep Learning”. In: *Procs. of the IEEE Int. Conference on Mobile Ad Hoc and Sensor Systems 2019*. 2019, pp. 476–484 (cit. on p. 246).

[733] K. Wüstefeld and F. Weichert. “An Automated Rapid Test for Viral Nanoparticles Based on Spatiotemporal Deep Learning”. In: *Procs. of IEEE SENSORS 2020*. Oct. 2020, pp. 1–4. DOI: 10.1109/SENSORS47125.2020.9278935 (cit. on pp. 27, 29–33). **SFB876-B2**

[734] Y. Xie et al. “Early lung cancer diagnostic biomarker discovery by machine learning methods”. In: *Translational oncology* 14.1 (2021), p. 100907 (cit. on p. 10).

[735] H. Xu and K. Larson. “Improving the efficiency of crowdsourcing contests”. In: *Procs. of the Int. Conference on Autononomous Agents and Multi-Agent Systems 2014*. 2014 (cit. on p. 298).

[736] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang. “Federated learning for healthcare informatics”. In: *Journal of Healthcare Informatics Research* 5.1 (2021), pp. 1–19 (cit. on p. 14).

[737] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. “How Powerful are Graph Neural Networks?” In: *Procs. of the Int. Conference on Learning Representations 2019*. 2019 (cit. on p. 89).

[738] G. Xue, J. Xu, H. Wu, and W. Lu. “Incentive Mechanism for Rational Miners in Bitcoin Mining Pool”. In: *Information Systems Frontiers* 23.10 (2021) (cit. on pp. 299, 302, 309).

[739] M. Yakout, L. Berti-Équille, and A. K. Elmagarmid. “Don’t be scared: use scalable automatic repairing with maximal likelihood and bounded changes”. In: *Procs. of the ACM SIGMOD Int. Conference on Management of Data 2013*. 2013, pp. 553–564 (cit. on p. 12).

[740] K. Yamanishi and J. Takeuchi. “A Unifying Framework for Detecting Outliers and Change Points from Non-stationary Time Series Data”. In: *Procs. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2002*. KDD ’02. New York, NY, USA: ACM, 2002 (cit. on p. 168).

[741] J. Yan, X. Yin, W. Lin, C. Deng, H. Zha, and X. Yang. “A Short Survey of Recent Advances in Graph Matching”. In: *Procs. of the ACM on Int. Conference on Multimedia Retrieval 2016*. Ed. by J. R. Kender, J. R. Smith, J. Luo, S. Boll, and W. H. Hsu. ACM, 2016, pp. 167–174. DOI: https://doi.org/10.1145/2911996.2912035 (cit. on p. 87).

[742] L. Yan et al. “An interpretable mortality prediction model for COVID-19 patients”. In: *Nature machine intelligence* 2.5 (2020), pp. 283–288 (cit. on p. 11).

[743] D. Yang, G. Xue, X. Fang, and J. Tang. “Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing”. In: *Procs. of ACM MobiCom 2012*. 2012 (cit. on pp. 298, 299).

[744] K. Yang, B. Huang, J. Stoyanovich, and S. Schelter. “Fairness-Aware Instrumentation of Preprocessing Pipelines for Machine Learning”. In: *Procs. of the Workshop on Human-In-the-Loop Data Analytics 2020*. 2020 (cit. on p. 13).

[745] Q. Yang, Y. Liu, T. Chen, and Y. Tong. “Federated machine learning: Concept and applications”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19 (cit. on p. 14).

[746] M. Yayla et al. “Nanoparticle Classification Using Frequency Domain Analysis on Resource-Limited Platforms”. In: *Sensors* 19 (2019). DOI = 10.3390/s19194138. URL: https://www.mdpi.com/1424-8220/19/19/4138 (cit. on pp. 30, 31). **SFB876-B2**

[747] Z. Ye, H. Xiong, and L. Wang. “Collecting Comprehensive Traffic Information Using Pavement Vibration Monitoring Data”. In: *Computer-Aided Civil and Infrastructure Engineering* 35.2 (Jan. 2020), pp. 134–149. URL: https://doi.org/10.1111/mice.12448 (cit. on p. 245).

[748] R. Yearby. “Racial disparities in health status and access to healthcare: the continuation of inequality in the United States due to structural racism”. In: *American Journal of Economics and Sociology* 77.3-4 (2018), pp. 1113–1152 (cit. on p. 18).

[749] J. Yoon, J. Jordon, and M. Schaar. “Gain: Missing data imputation using generative adversarial nets”. In: *Procs. of the Int. Conference on Machine Learning 2018*. PMLR. 2018, pp. 5689–5698 (cit. on p. 13).

[750] F. X. Yu, S. Kumar, T. Jebara, and S. Chang. “On Learning with Label Proportions”. In: *arXiv: Computing Research Repository* (2014). DOI: arXiv:1402.5902 (cit. on pp. 142–146).

[751] F. X. Y. Yu, D. Liu, S. Kumar, T. Jebara, and S. Chang. “$\propto$SVM for Learning with Label Proportions”. In: *Procs. of the Int. Conference on Machine Learning 2013*. 2013, pp. 504–512 (cit. on p. 140).

[752] P. Yuan, A. Rezvan, X. Li, N. Varadarajan, and H. Van Nguyen. “Phasetime: Deep Learning Approach to Detect Nuclei in Time Lapse Phase Images”. In: *Journal of Clinical Medicine* 8.8 (2019) (cit. on p. 32).

[753] L. Z. Yunpu Ma Volker Tresp and Y. Wang. “Variational Quantum Circuit Model for Knowledge Graph Embedding”. In: *Advanced Quantum Technologies* (2019). DOI: DOI:10.1002/qute.201800078 (cit. on p. 388).

[754] A. Zeltins. “Construction and Characterization of Virus-Like Particles: A Review”. In: *Molecular Biotechnology* 53.1 (2013), pp. 92–107. URL: %3CGo%20to%20ISI%3E://WOS:000312640400011 (cit. on p. 23).

[755] Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou. “Comparing Stars: On Approximating Graph Edit Distance”. In: *Procs. VLDB Endowment* 2.1 (Aug. 2009), pp. 25–36. URL: http://dx.doi.org/10.14778/1687627.1687631 (cit. on pp. 93, 94).

[756] J. Zentgraf and S. Rahmann. “Fast lightweight accurate xenograft sorting”. In: *Algorithms for Molecular Biology* 16.1 (2021), p. 2. URL: https://doi.org/10.1186/s13015-021-00181-w (cit. on p. 46). **SFB876-C1**

[757] J. Zentgraf, H. Timm, and S. Rahmann. “Cost-optimal assignment of elements in genome-scale multi-way bucketed Cuckoo hash tables”. In: *Procs. of the Symposium on Algorithm Engineering and Experiments 2020*. SIAM, 2020, pp. 186–198. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611976007.15 (cit. on p. 46). **SFB876-C1**

[758] A. Zhang, S. Song, J. Wang, and P. S. Yu. “Time series data cleaning: From anomaly detection to anomaly repairing”. In: *Procs. of the VLDB Endowment* 10.10 (2017), pp. 1046–1057 (cit. on p. 12).

[759] K. Zhang, M. Hutter, and H. Jin. “A new local distance-based outlier detection approach for scattered real-world data”. In: *Procs. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining 2009*. Springer. 2009, pp. 813–822 (cit. on p. 14).

[760] Q. Zhang, Y. Liu, G. Liu, G. Zhao, Z. Qu, and W. Yang. "An automatic diagnostic system based on deep learning, to diagnose hyperlipidemia". In: *Diabetes, metabolic syndrome and obesity: targets and therapy* 12 (2019), p. 637 (cit. on p. 6).

[761] T. Zhang and R. Ma. "Identify the cells' nuclei based on the deep learning neural network". In: *arXiv: Computing Research Repository* (2019). DOI: arXiv:1911.09830 (cit. on p. 32).

[762] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li. "Free market of crowdsourcing: Incentive mechanism design for mobile sensing". In: *IEEE Transactions on Parallel and Distributed Systems* 25.12 (2014) (cit. on pp. 298, 299, 302).

[763] X. Zhang et al. "Incentives for Mobile Crowd Sensing: A Survey". In: *IEEE Communications Surveys and Tutorials* 18.1 (2015) (cit. on p. 298).

[764] X. Zhang, C. Gu, and J. Lin. "Support vector machines for anomaly detection". In: *Procs. of the World Congress on Intelligent Control and Automation 2006*. Vol. 1. IEEE. 2006, pp. 2594–2598 (cit. on pp. 9, 14).

[765] X. Zhang, X. Lu, S. Wang, W. Wang, and W. Li. "A multi-sensor based online tool condition monitoring system for milling process". In: *Procedia CIRP* 72 (2018), pp. 1136–1141 (cit. on p. 158).

[766] Y. Zhang, Y. Fang, and S. Zhong. "Incentive mechanism design for smartphone crowdsensing". In: *Procs. of the IEEE Int. Conference on Big Data and Cloud Computing 2015*. 2015 (cit. on pp. 298, 299, 302).

[767] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* (2017) (cit. on p. 130).

[768] X. Zhao, C. Xiao, X. Lin, W. Wang, and Y. Ishikawa. "Efficient processing of graph similarity queries with edit distance constraints". In: *VLDB J.* 22.6 (2013), pp. 727–752 (cit. on pp. 93, 94).

[769] X. Zhao, C. Xiao, X. Lin, W. Zhang, and Y. Wang. "Efficient structure similarity searches: a partition-based approach". In: *VLDB – The International Journal on Very Large Data Bases* 27.1 (2018), pp. 53–78 (cit. on pp. 93, 94).

[770] W. Zheng, L. Zou, X. Lian, D. Wang, and D. Zhao. "Efficient Graph Similarity Search Over Large Graph Databases". In: *IEEE Transactions on Knowledge and Data Engineering* 27.4 (2015), pp. 964–978 (cit. on p. 93).

[771] R. Zou and W. Wang. "U-CIMAN: Uncover Spectrum and User Information in LTE Mobile Access Networks". In: *Procs. of the IEEE Conference on Computer Communications 2020*. 2020, pp. 1459–1468 (cit. on p. 347).

[772] A. Zybin, Y. Kuritsyn, E. Gurevich, V. Temchura, K. Überla, and K. Niemax. "Real-time Detection of Single Immobilized Nanoparticles by Surface Plasmon Resonance Imaging". In: *Plasmonics* 5 (2009), pp. 31–35 (cit. on p. 24).

[773] A. Zybin, V. Shpacovitch, J. Skolnik, and R. Hergenröder. "Optimal conditions for SPR-imaging of nano-objects". In: *Sensors and Actuators B: Chemical* 239 (2017), pp. 338–342 (cit. on p. 25). **SFB876-B2**

[774] N. Zygouras, N. Zacheilas, V. Kalogeraki, D. Kinane, and D. Gunopulos. "Insights on a Scalable and Dynamic Traffic Management System". In: *Procs. of the Int. Conference on Extending Database Technology 2015*. 2015, pp. 653–664 (cit. on p. 235).

Index

3D Euclidean positioning error, 382
5G New Radio (NR), 385
5G networks, 224, 318, 363, 377
5G positioning, 388
6G networks, 329, 387

Active Learning (AL), 160
Age of Information (AoI), 333
Aggregate Output Classification Kernel k-Means (AOC-KK), 141
Akaike's Information Criterion (AIC), 325
Algebraic Equation Set (AES), 198
Angle-based positioning, 388
Angle-of-Arrival (AoA), 383
Angle-of-Departure (AoD), 383
Antenna
– directivity, 376
– Phased Array (PAA), 376
Anticipatory mobile networking, 275, 332
Attenuation patterns, 241
Augemented Reality (AR), 355
Autoencoder, 16
Automated vehicles (AVs), 293
AutoRegressive Integrated Moving Average (ARIMA), 325, 366

Background, 27, 34
Backpropagation, 31
Bagging, 125
Bayes
– naive Bayes classifier, 124
– optimal Bayes classifier, 143
– theorem, 58
Bayesian network, 228
Beam
– alignment, 375
– management, 385
– sweeping, 376
– tracking, 375
Bearing-based mmWave positioning, 383
Better Approach to Mobile Ad-hoc Networking (B.A.T.M.A.N.), 273
Bias, 18, 49, 78, 166, 169–172
Binary tree, 67
Biomarker discovery, 9
Black spot, 338
Blackspot-Aware Contextual Bandit (BS-CB), 331
Block Error Rates (BLER), 329
Boosting, 9, 71, 125

Cellular automaton model, 291, 293
Central Processing Unit (CPU), 35, 45, 46
Central Unit (CU), 385
Channel Impulse Response (CIR), 241
Channel phase tracking, 388
Channel Quality Indicator (CQI), 334
Channel State Information (CSI), 241
Channel-Aware Transmission (CAT), 331
Clock synchronization, 184
Clustering, 82, 86, 127, 161
Commercial-Off-The-Shelf (COTS), 356
Communicating Automated Vehicles (CAVs), 293
Communication
– Enhanced Machine Type (eMTC), 261
– massive Machine Type (mMTC), 261
– massive Machine Type Communication (mMTC), 313, 331, 355
Competition-free channel access, 189
Confidentiality, 15
– enforcement, 394
– requirement, 393
Constrained Application Protocol (CoAP), 266
Coresets, 59
Critical infrastructure communications, 358
Cross validation, 81, 130, 152, 155
Cross-validation, 80
Crowdsensing, 297, 331
Cyber-Physical System (CPS), 104, 180, 241
Cyber-physical twin, 109

Data
– acquisition, 195
– stream algorithm, 66
Data-Driven Model-Predictive Communication, 323
Data-Driven Network Simulation (DDNS), 338
Database search, 86, 88, 93

Open Access. © 2023 the author(s), published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

https://doi.org/10.1515/9783110785968-008

Decision tree, 124
Deep Boltzmann Trees (DBTs), 254
Deoxyribonucleic acid (DNA), 23
– fragment, 45
– sequencing, 43, 49
– transcription, 85
Detect-And-Avoid (DAA), 315
Detector, 29, 30, 32
Differential privacy, 231
Digital twin, 108
Dilution Of Precision (DOP), 384
Distributed machine learning, 137
Distributed Unit (DU), 385
Downlink Control Information (DCI), 346
Duty cycle, 315
Dynamic transit routing, 233

Embedded device, 37
Embedded system, 35
End-to-end system architecture, 384
Energy, 35
– consumption, 21, 22, 35, 36
– efficiency, 35, 185
– harvesting, 195
– saving, 35
enhanced Mobile Broadband (eMBB), 355
Ensembles, 9, 127, 162, 165, 169, 170, 172
– resampling-based, 125
Error
– mean squared, 164, 169
– measures, 161
Error Vector Magnitude (EVM), 380
Euclidean distance, 69, 168
Evolutionary algorithms, 147
Exponential families, 229
Extended Kalman Filter (EKF), 187

Fast Analysis of LTE CONtrol channels (FALCON), 344
Feature selection, 9, 58, 59, 71, 133
Field-Programmable Gate Array (FPGA), 41
Floating Car Data (FCD), 356

Gaussian processes, 228
Genetic
– analysis, 43
– optimization, 34
Genetic programming algorithm, 123
Global Navigation Satellite Systems (GNSS), 181, 242
Graphical models, 142, 228
Graphics Processing Unit (GPU), 35, 45

Hashing
– Cuckoo, 46
Hierarchical Token Bucket (HTB), 356
High accuracy 3D motion tracking, 388
Human-driven Vehicles (HVs), 293

Incentive mechanisms, 297
– data-reverse-auction, 299
– dynamic relative, 304
– static uniform, 304
– task-reverse-auction, 299
Industry 4.0, 103, 158, 179, 182, 196, 212
Inference, 65
– proofness, 391
Information
– communicating, 392
– exploiting, 392
– keeping, 392
– system, 393
Integer linear programming, 88, 95
Intelligent Transportation Systems (ITS), 241, 272, 356
Internet of Things (IoT), 137, 213, 261
– cellular, 261
– Narrowband (NB-IoT), 261

Joint Communication And Sensing (JCAS), 387

k-Means
– Fast, 151
k-Nearest Neighbour (kNN), 124
Kernel functions, 228
– graph, 87, 89
– Radial Basis Function (RBF), 335
Key Performance Indicator (KPI), 320, 354
Knowledge
– a priori, 36, 40, 52, 58, 395
– background, 32, 86, 95, 395
– epistemic state, 394
– inferred, 396

L1 norm, 59
L2 norm, 59
Label
– counts, 232
– proportions, 231
Laplace distribution, 232
Latent variable model, 48, 49, 53
Learning
– semi-supervised, 137

– supervised, 123, 137, 242, 309
– unsupervised, 124, 137, 309
Learning from Label Proportions (LLP), 136, 137, 146, 231, 233
Learning from Label Proportions by Clustering (LLPC), 147
Learning tasks, 173, 175
Lee model, 290, 293
LIghtweight Machine Learning for IoT Systems with resource limitations (LIMITS), 277, 335
Line-Of-Sight (LOS), 279, 380
– obstructed, 385
Link Expiry Time (LET), 276
Link quality, 269
Listen-Before-Talk (LBT), 315
Logistics, 110
Long Short Term Memory (LSTM), 326
Long-Range Wide-Area Network (LoRaWAN), 262, 315
Low-Power Wide Area Network (LPWAN), 313
Low-Power Wide-Area Network (LPWAN), 261, 262

M5 regression tree, 335
Machine Learning-CAT (ML-CAT), 331
Machine to Machine (M2M), 265
Machine-Type Communications (MTC), 329
Management And Network Orchestration (MANO), 356, 385
Markov chains, 50, 65
Markov Random Fields (MRFs), 229
Maturity index, 106
Maximum likelihood, 80
Maximum Power Point (MPP), 196
Median filtering, 29
Medical data analysis, 22
Medium Access Control (MAC), 315
Memory, 66, 74, 77
– utilization, 254
Merge & Reduce, 58, 59, 66
Message Queuing Telemetry Transport (MQTT), 220, 265
Message Queuing Telemetry Transport for Sensor Networks (MQTT-SN), 266
Millimeter wave (mmWave)
– communication, 375, 389
– network coverage, 385
– pencil beam, 375
– radio links, 378
Min-hashing, 47, 88, 89, 93
Mixture model, 46
Mobile Ad-hoc Networks (MANETs), 272
Mobile radio networks, 385
Mobile system, 21
Mobility-predictive routing, 274
Motion Capture (MoCap) system, 217
Multi-modal trip planning, 233
Multilayer perceptrons, 162
Nagel-Schreckenberg model, 227, 287
Nash equilibrium, 299
Network
– coverage, 263, 389
– slicing, 363, 368
Network Data Analytic Function (NWDAF), 340
Network Function Virtualization (NFV), 355
Network Interface Card (NIC), 362
Neural Network
– Convolutional Neural Networks (CNN), 6, 8, 32, 142
– Fully Convolutional Networks (FCN), 176
– Generative Adversarial Networks (GAN), 38, 40
– Recurrent Neural Networks (RNN), 326
– Residual Networks model (ResNet), 32

Offloading, 35
Operating system, 362
Optimized Link State Routing (OLSR), 273

PAMONO, 21, 31, 33, 36
Parallelism, 45
Particle, 23, 27, 30–32
– detection, 30
Passive reflectors, 385
Pearson correlation
– coefficient, 97, 127
Personalized therapy, 43
Pervasive intelligence, 332
Photovoltaics, 195
Physical Downlink Control CHannel (PDCCH), 346
Physical layer Service Data Unit (PSDU), 190

Poisson Dependency Network (PDN), 230, 289
Policy Optimizer for Exponential Models (POEM), 236
Power
– consumption, 190
Precision Time Protocol (PTP), 358
Predictive Ad-hoc Routing Fueled by Reinforcement Learning and Trajectory knowledge (PARRoT), 273
Private networks, 388
Proportional hazards regression, 63
Protein
– complex, 85, 86
– interactions, 88, 95
Proximal Policy Optimization (PPO), 223
Proximity forest, 254
Pruning, 127

Q-learning, 275
Q-routing, 274
Quality of Service (QoS), 262, 332, 365, 385

Radio
– based sensing, 388
– fingerprints, 246
– Radio Network Temporary Identifier (RNTI), 343
– smart radio environments, 385
– Software-Defined Radio (SDR), 320, 345
  – based mmWave platform, 375
Random forest, 31, 171, 254, 277, 335
Random walk Metropolis-Hastings method, 64
Read mapper, 45, 49
Real-time capability, 21
Real-Time Clock (RTC), 186
Real-Time Localization System (RTLS), 184
Reconfigurable Intelligent Surface (RIS), 385
– placement, 386
Regression, 71, 123, 162
– Bayesian, 58
– Gaussian process regression, 80
– generalized linear regression models, 66
– LASSO, 9, 80, 81
– linear, 60, 65, 67
– logic, 61
– ordinary least squares, 59
– Penalized regression models, 79
– Poisson tree, 230
– proportional hazards, 62, 76, 79
Regularization, 59, 63, 130, 161
Reinforcement learning, 127, 221, 272, 273, 283
Reinforcement Learning-CAT (RL-CAT), 331
Resource
– optimization, 22, 35
Resource Block (RB), 346
Resource Element (RE), 345
Resource-constrained, 182
Ribonucleic acid (RNA), 23
Risk classification, 43
Road-side Unit (RSU), 273
Robot Operating System (ROS), 184, 217
Root Mean Square Error (RMSE), 327
Routing
– algorithm, 272
– protocols, 272–274

Sampling, 50, 81
– informed, 162
– Markov-Chain-Monte-Carlo (MCMC), 64, 68, 139
– random, 165
Scalability, 58, 65, 188
Scheduling, 185, 364
Seasonal and Trend decomposition using LOESS (STL), 324
Security
– availability, 397
– confidentiality, 397
– rational reasoner, 396
Security mechanism, 395
– overhead, 409
Security policy, 396
– permission, 394
– prohibition, 395
Self-Organizing Maps (SOM), 133
Sequencing coverage, 47
Sequential Minimal Optimization (SMO), 335
Short-Range Device (SRD), 315
Signal
– background separation, 29
– quality measurements, 378
– Received Signal Strength Indicator (RSSI), 242, 318
– Reference Signal Received Power (RSRP), 334
– Reference Signal Received Quality (RSRQ), 334
– Signal-to-Interference-plus-Noise Ratio (SINR), 332
– Signal-to-Noise Ratio (SNR), 32, 350

Similarity
– cosine, 97, 98
– graph edit, 85, 92, 93
– Jaccard, 87, 88
– Weisfeiler-Leman, 88, 91, 93
Similarity measures, 86, 89
Simulation of Urban MObility (SUMO), 239
Sketches, 59, 89
Slice-Aware Machine Learning-Based Ultra-Reliable Scheduling (SAMUS), 365
Slice-Aware Machine Learning-based Ultra-Reliable Scheduling (SAMUS), 354
Smart Grid communications, 354
Software-Defined Networking (SDN), 355, 385
Spatio-temporal deep learning, 31
Spectral efficiency, 261
Speedup, 36
Stackelberg game, 299
Sub-sampling, 81
Support Vector Data Description (SVDD), 125
Support Vector Machine (SVM), 124, 133, 254, 277, 335
– Regression (SVR), 140, 158
Surface Plasmon Resonance (SPR), 24
Surrogate model, 71, 77
Synchronization, 187

Template matching, 31, 32
Time Of Arrival (TOA), 181
Time series data, 27, 119, 126, 133, 161, 168, 324
Time-Difference Of Arrival (TDOA), 184
Time-on-Air (ToA), 318
Timing Advance (TA), 334
Transfer learning, 32
Transit networks, 233
Transmission Time Interval (TTI), 365
Tumor, 43, 53, 64
– primary, 43
– relapse, 43
Two-Way Ranging (TWR), 190

Ultra-Low Power (ULP), 195
ultra-Reliable Low Latency Communication (uRLLC), 354, 356
Ultra-Wideband (UWB), 181
Unmanned Aerial Vehicles (UAV), 112, 184, 212, 272, 378
User Datagram Protocol (UDP), 358
User Equipment (UE), 344, 365
User positioning, 388

Variable Importance Measure, 59
Variance, 78, 166, 169, 170, 172
Variant calling, 48
Virtual Reality (VR), 355
Virus, 21

Waikato Environment for Knowledge Analysis (WEKA), 335
Weighted Euclidean distance, 151
Wireless Sensor Network (WSN), 242
Workflow management system, 44

Xenograft sorting, 46

List of Contributors

Editors

Morik, Katharina, Prof. Dr. TU Dortmund University, Department of Computer Science, Head of the chair for Artificial Intelligence & Speaker of the Collaborative Research Center 876 on Providing Information by Resource-Constrained Data Analysis, katharina.morik@tu-dortmund.de

Wietfeld, Christian, Prof. Dr. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Head of the chair for Communication Networks & Co-Speaker of the Collaborative Research Center 876, christian.wietfeld@tu-dortmund.de

Rahnenführer, Jörg, Prof. Dr. TU Dortmund University, Department of Statistics, Head of the group for Statistical Methods in Genetics and Chemometrics, CRC 876, rahnenfuehrer@statistik.tu-dortmund.de

Contributors

Arendt, Christian, M. Sc. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876,

Awasthi, Shrutarv, M. Sc. TU Dortmund University, Faculty of Mechanical Engineering, Chair of Materials Handling and Warehousing, FLW, CRC 876, shrutarv.awasthi@tu-dortmund.de

Bektas, Caner, M. Sc. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, caner.bektas@tu-dortmund.de

Biskup, Joachim, Prof. Dr. TU Dortmund University, Department of Computer Sciences, Databases and Information Systems Group, CRC 876, joachim.biskup@cs.tu-dortmund.de

Borgwardt, Karsten, Prof. Dr. ETH Zurich, Department of Biosystems Science and Engineering, Machine Learning & Computational Biology Lab, karsten.borgwardt@bsse.ethz.ch

Böcker, Stefan, Dipl.- Ing. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876, stefan.boecker@tu-dortmund.de

Büscher, Jan, M. Sc. TU Dortmund University, Faculty of Mechanical Engineering, Institute of Production Systems (IPS), CRC 876, jan.buescher@ips.tu-dortmund.de

Cheng, Liang, Dr. University of Toledo, Electrical Engineering & Computer Science Department, EECS Department Chair & Professor, liang.cheng@utoledo.edu

Deuse, Jochen, Univ.-Prof. Dr.-Ing. TU Dortmund University, Faculty of Mechanical Engineering, Institute of Production Systems (IPS), CRC 876, jochen.deuse@ips.tu-dortmund.de

Ding, Zeyu, M. Sc. TU Dortmund University, Department of Statistics, CRC 876, zeyu.ding@tu-dortmund.de

Open Access. © 2023 the author(s), published by De Gruyter. CC BY This work is licensed under the Creative Commons Attribution 4.0 International License.
https://doi.org/10.1515/9783110785982-009

Finkeldey, Felix, M. Sc. TU Dortmund University, Department of Computer Sciences, Chair for Software Engineering, CRC 876, felix.finkeldey@tu-dortmund.de

Gramse, Nils, M. Sc. TU Dortmund University, Faculty of Mechanical Engineering, Chair of Materials Handling and Warehousing, FLW, nils.gramse@tu-dortmund.de

Haferkamp, Marcus, M. Sc. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876, marcus.haferkamp@tu-dortmund.de

Heimann, Karsten, M. Sc. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876, karsten.heimann@tu-dortmund.de

Hergenröder, Roland, Dr. Leibniz-Institut für Analytische Wissenschaften - ISAS - e.V., CRC 876, roland.hergenroeder@isas.de

ten Hompel, Michael, Prof. Dr. Dr. h. c. Fraunhofer-Institut für Materialfluss und Logistik IML, CRC 876, michael.tenhompel@tu-dortmund.de

Ickstadt, Katja, Prof. Dr. Department of Statistics, Mathematical Statistics with Applications in Biometrics, CRC 876, ickstadt@statistik.tu-dortmund.de

Jörke, Pascal, M. Sc. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876, pascal.joerke@tu-dortmund.de

Jutzeler, Catherine, Prof. Dr. ETH Zurich, Department of Health Sciences and Technology, catherine.jutzeler@hest.ethz.ch

Köster, Johannes, Dr. Duisburg-Essen University, University Hospital Essen, Institute of Human Genetics, CRC 876, johannes.koester@uk-essen.de

Kriege, Nils, Ass.-Prof. Dipl.-Inf. Dr. University of Vienna, Research Group Data Mining and Machine Learning, CRC 876, nils.kriege@univie.ac.at

Krieger, Cedrik, M. Sc. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876, cedrik.schueler@tu-dortmund.de

Kurtz, Fabian, Dr.-Ing. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, fabian.kurtz@tu-dortmund.de

Lang, Michel, Dr. TU Dortmund University, Department of Statistics, Statistical methods in Genetics and Chemometrics, CRC 876, lang@statistik.tu-dortmund.de

Liebig, Thomas, Juniorprof. Dr. TU Dortmund University, Department of Computer Sciences, Chair for Artificial Intelligence, CRC 876, thomas.liebig@cs.tu-dortmund.de

Machado, Maximilian, M. Sc. Lehigh University, Department of Electrical and Computer Engineering, maximillian.k.machado@gmail.com

Masoudinejad, Mojtaba, Dr.-Ing. TU Dortmund University, Department of Computer Science, Design Automation for Embedded Systems Group, CRC 876, mojtaba.masoudinejad@tu-dortmund.de

Munteanu, Alexander, Dr. TU Dortmund University, Dortmund Data Science Center, Faculties of Statistics and Computer Science, CRC 876, alexander.munteanu@tu-dortmund.de

Mutzel, Petra, Prof. Dr. Bonn University, Institute of Computer Science, Computer Science I, CRC 876, petra.mutzel@cs.uni-bonn.de

Overbeck, Dennis, M. Sc. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, dennis.overbeck@tu-dortmund.de

Panusch, Thorben, M. Sc. TU Dortmund University, Department of Computer Science, CRC 876, thorben.panusch@ips.tu-dortmund.de

Rahmann, Sven, Prof. Dr. Saarland University, Center for Bioinformatics, CRC 876, sven.rahmann@tu-dortmund.de

Ran, Ran, M. Sc. Lehigh University, Department of Electrical and Computer Engineering, rar418@lehigh.edu

Richter, Jakob, Dr. TU Dortmund University, Department of Statistics, Statistical methods in Genetics and Chemometrics, CRC 876, richter@statistik.tu-dortmund.de

Roidl, Moritz, Dipl.-Inform. TU Dortmund University, Faculty of Mechanical Engineering, Chair of Materials Handling and Warehousing, FLW, CRC 876, moritz.roidl@tu-dortmund.de

Saadallah, Amal, M. Sc. TU Dortmund University, Department of Computer Sciences, Chair for Artificial Intelligence, CRC 876, amal.saadallah@cs.tu-dortmund.de

Schramm, Alexander, Prof. Dr. University Hospital Essen, Department of Medical Oncology, CRC 876, alexander.schramm@uk-essen.de

Schreckenberg, Michael, Prof. Dr. University of Duisburg Essen, Faculty for Physics, Physics for Transport and Traffic, CRC 876, michael.schreckenberg@uni-due.de

Shpacovitch, Victoria, Dr. Leibniz-Institut für Analytische Wissenschaften - ISAS - e.V., CRC 876, victoria.shpacovitch@isas.de

Sliwa, Benjamin, Dr.-Ing. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876, benjamin.sliwa@tu-dortmund.de

Stolpe, Marco, Dr. TU Dortmund University, Department of Computer Sciences, Chair for Artificial Intelligence, CRC 876, marco.stolpe@tu-dortmund.de

Stöcker, Bianca, M. Sc. Duisburg-Essen University, University Hospital Essen, Institute of Human Genetics, CRC 876, bianca.stoecker@uni-due.de

Tiemann, Janis, Dr.-Ing. TU Dortmund University, Faculty of Electrical Engineering and Information Technology, Communication Networks Institute, CRC 876, janis.tiemann@tu-dortmund.de

Vranken, Tim, M. Sc. University of Duisburg Essen, Faculty for Physics, Theoretical Physics, CRC 876, tim.vranken@uni-due.de

Weichert, Frank, Priv.-Doz. Dr. TU Dortmund University, Department of Computer Science, Intelligent Sensing in Computer Graphics, CRC 876,frank.weichert@tu-dortmund.de

Wiederkehr, Petra, Prof. Dr.-Ing. TU Dortmund University, Department of Computer Sciences, Chair for Software Engineering, CRC 876, petra.wiederkehr@tu-dortmund.de

Wüstefeld, Konstantin, M. Sc. TU Dortmund University, Department of Computer Sciences, CRC 876, konstantin.wuestefeld@tu-dortmund.de

Technical Editors

Becker, Andreas, Dr. TU Dortmund University, Department of Computer Science, CRC 876, andreas3.becker@tu-dortmund.de

Buß, Jens, Dr. TU Dortmund University, Department of Computer Science, Managing Director of the Collaborative Research Center 876, jens.buss@tu-dortmund.de

Acknowledgment

Part of the work on this book is the result of research of the Collaborative Research Center 876 "Providing Information by Resource-Constrained Analysis", which was funded from 2011–2022 by the Deutsche Forschungsgemeinschaft (DFG) under DFG project number 124020371, see: https://gepris.dfg.de/gepris/projekt/124020371?language=en.