Research Article

Atiaf A. Rawi*, Murtada K. Elbashir, and Awadallah M. Ahmed

# Deep learning models for multilabel ECG abnormalities classification: A comparative study using TPE optimization

https://doi.org/10.1515/jisys-2023-0002
received January 07, 2023; accepted March 06, 2023

**Abstract:** The problem addressed in this study is the limitations of previous works that considered electrocardiogram (ECG) classification as a multiclass problem, despite many abnormalities being diagnosed simultaneously in real life, making it a multilabel classification problem. The aim of the study is to test the effectiveness of deep learning (DL)-based methods (Inception, MobileNet, LeNet, AlexNet, VGG16, and ResNet50) using three large 12-lead ECG datasets to overcome this limitation. The define-by-run technique is used to build the most efficient DL model using the tree-structured Parzen estimator (TPE) algorithm. Results show that the proposed methods achieve high accuracy and precision in classifying ECG abnormalities for large datasets, with the best results being 97.89% accuracy and 90.83% precision for the Ningbo dataset, classifying 42 classes for the Inception model; 96.53% accuracy and 85.67% precision for the PTB-XL dataset, classifying 24 classes for the Alex net model; and 95.02% accuracy and 70.71% precision for the Georgia dataset, classifying 23 classes for the Alex net model. The best results achieved for the optimum model that was proposed by the define-by-run technique were 97.33% accuracy and 97.71% precision for the Ningbo dataset, classifying 42 classes; 96.60% accuracy and 83.66% precision for the PTB-XL dataset, classifying 24 classes; and 94.32% accuracy and 66.97% precision for the Georgia dataset, classifying 23 classes. The proposed DL-based methods using the TPE algorithm provide accurate results for multilabel classification of ECG abnormalities, improving the diagnostic accuracy of heart conditions.

**Keywords:** heart disease, ECG, 12-lead ECG signal, deep learning models, PTB-XL, tree-structured Parzen estimator algorithm

## 1 Introduction

Heart disease remains the leading cause of death and places a significant burden on global healthcare systems [1]. Electrocardiogram (ECG) is a reliable, non-invasive, and widely used approach for monitoring heart function, collecting wave information from 12 electrodes on the chest, ankles, and wrist [2]. With approximately 300 million ECGs recorded worldwide each year, early and accurate ECG diagnosis is critical for saving lives [3]. However, incorrect interpretation of ECGs can lead to incorrect clinical decisions and adverse outcomes [4]. Therefore, developing a computer-aided diagnostic system to interpret ECG signals is crucial, particularly in countries where expert cardiologists are in short supply, and heart abnormalities

* **Corresponding author: Atiaf A. Rawi,** Department of Computer Sciences, Faculty of Mathematical and Computer Sciences, Gezira University, P.O. Box 20, Wad Madani 21111, Sudan, e-mail: atiaf.ayal88@gmail.com
**Murtada K. Elbashir:** Department of Information Systems, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Saudi Arabia, e-mail: mkelfaki@ju.edu.sa
**Awadallah M. Ahmed:** Department of Computer Sciences, Faculty of Mathematical and Computer Sciences, Gezira University, Wad Madani 21111, Sudan, e-mail: awadallah@uofg.edu.sd

have similarities [5]. Researchers have investigated traditional and deep learning (DL)-based methods for building an automatic and accurate ECG diagnosis system, utilizing open-source ECG datasets such as the Chapman ECG dataset [6], the Georgia 12-lead ECG Challenge Dataset [7], and the PTB-XL dataset [8]. Traditional methods require expert feature engineering and are time-consuming, whereas DL-based methods use convolutional neural networks (CNNs) to handle feature extraction efficiently, resulting in end-to-end models. Figure 1 illustrates the difference between traditional and DL-based ECG classification methods. Ultimately, DL-based methods offer more practical and efficient solutions to ECG classification, overcoming domain knowledge and data quality limitations of traditional methods [9].

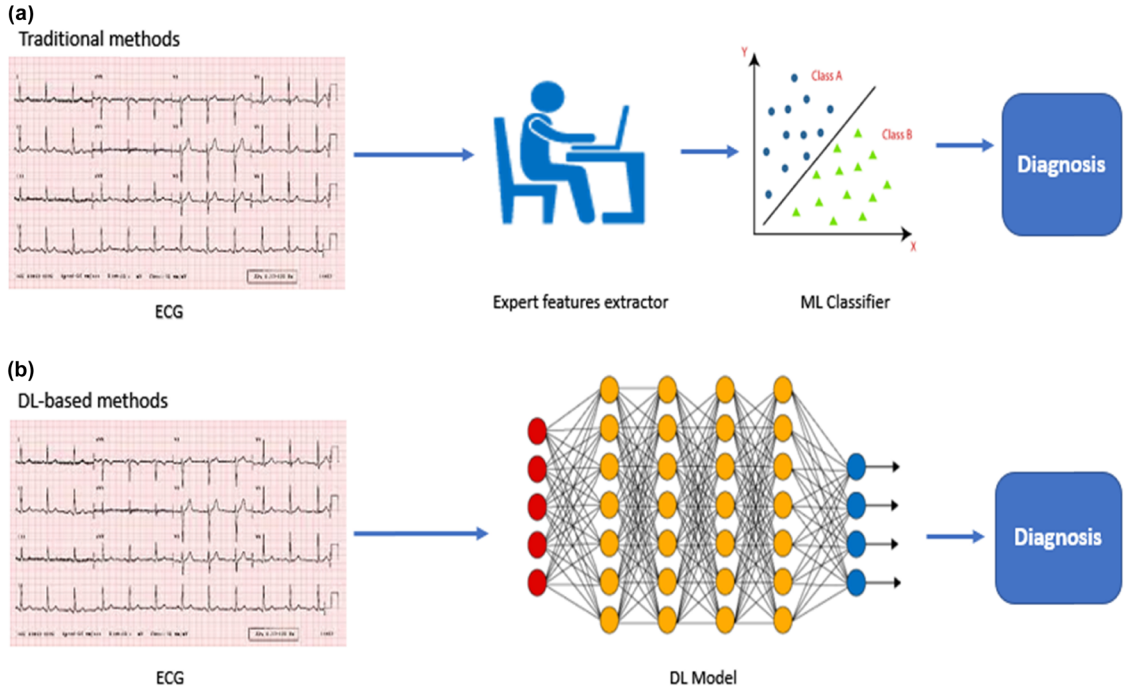


**Figure 1:** Comparison between (a) traditional and (b) DL-based methods for ECG signals classification [Original].

This article presents a comparative study of DL-based methods for multilabel ECG abnormalities classification using the tree-structured Parzen estimator (TPE) algorithm for model optimization. The proposed method can diagnose ECG records with multiple abnormalities in the same record, which is essential in real-life scenarios. The study used three large 12-lead ECG datasets with a varying number of classes to test the generalization ability of the proposed method. The experiments show that the proposed method outperforms traditional methods and achieves high accuracy and precision rates for all three datasets. Furthermore, the proposed model is lightweight and can be implemented on limited-resource edge devices, making it suitable for building real-time heart disease diagnosis systems. The following are the contributions of this work:

1. A CNN model is introduced using a define-by-run technique through the TPE optimization algorithm to select the best number of CNN layers and kernel size, among other hyperparameters, to build the optimum model. This technique is novel and can improve the efficiency of DL models.
2. The proposed method can diagnose ECG records with multiple abnormalities in the same ECG record, as it uses a multilabel classification method, which is more accurate and relevant to real-world scenarios.
3. Three different datasets were used to test the generalization ability of the proposed method with the largest number of classes (42, 24, and 23). This shows that the model is robust and can perform well on different datasets.
4. The proposed model is lightweight and can be implemented on limited-resource edge devices to build a real-time heart disease diagnosis system, which can help in saving lives, especially in countries with a shortage of expert cardiologists.

The article is structured as follows: Section 2 provides a summary of previous related work. Section 3 introduces the dataset and the premise of the suggested methods. Section 4 details the experiments and their results. Comprehensive discussions are offered in Section 5, followed by the conclusion in Section 6.

## 2 Related work

Recently, DL models achieved promising results in many fields, such as computer vision [10] and natural language processing [11]. Additionally, DL techniques are utilized to assist diagnoses in pathology, ophthalmology, radiography, and dermatology [12]. In addition, DL models have recently been utilized to evaluate ECG data for various applications, including disease identification, annotation, and sleep staging [13]. Chen et al. [14] proposed a CNN model for detecting nine types of cardiac arrhythmia abnormalities using a 12-lead dataset provided by the China Physiological Signal Challenge (CPSC) 2018. The proposed method achieved an $F1$ score of 0.84 and ranked first in the competition. Another model is presented in ref. [15] for classifying atrial fibrillation, non-atrial fibrillation, and normal ECG signals using the Physionet Challenge 2017 dataset. Their method achieved average $F1$ scores of 0.79, 0.77, and 0.91, respectively, classifying atrial fibrillation, non-atrial fibrillation, and normal. In ref. [16], a deep neural network model is proposed using a single-lead ECG dataset to classify 12 signal classes. The average area under the receiver operating characteristics curve achieved is 0.97. A DL model is proposed by He et al. [17] using residual CNN and long short-term memory (LSTM) layers to classify nine classes of ECG signals using the CPSC dataset, and they achieved an overall $F1$ score of 0.806. In the study of Strodthoff et al. [18], many algorithms were tested, and the best results were achieved using ResNet and Inception-based architectures with the PTB-XL dataset to classify nine ECG signals. The study proposed by Ullah et al. [19] transforms the 1D ECG signals into spectral images using the Fourier transformation and then uses a DL model to classify eight classes of abnormalities. The MIT-BIH single-lead arrhythmia dataset was used, and the classification accuracy was 99.11%. The study by Zhang et al. [20] proposed a DL model for classifying nine cardiac arrhythmias using 12-lead ECG signals, reaching an average accuracy of 96.6%. Furthermore, they concluded that using single-lead signals dropped the accuracy to 11.8%. The study by He et al. [21] used 2,184 normal records and 3,201 records containing five heart abnormalities from the PTB-XL dataset to train a DL model based on CNN layers with the attention mechanism. The dataset was preprocessed before training using wavelet transformation and the Pan–Tompkins algorithm [22], resulting in an accuracy of 99.63%.

It is noteworthy that all the studies mentioned previously are limited to a maximum of nine classes of heart abnormalities. Consequently, they classify ECG records assuming the presence of only one type of abnormality. However, considering that most ECG datasets contain more than one abnormality in the same record, this makes the classification a multilabel classification problem rather than a multiclass problem. In this article, to address this limitation, we introduce end-to-end DL-based models that use 12-lead ECG waveform data without a feature engineering process to detect most classes of heart abnormalities available in the datasets.

## 3 Materials and methods

This study aims to build a DL-based model to classify the ECG signals. In contrast to most previous works, which considered classifying the ECG signals as multiclass classification, this study uses multilabel classification, considering the presence of more than one heart abnormality in the same ECG record simultaneously. The tree of Parzen estimator (TPE) is utilized as an optimization algorithm to build the optimal CNN model by optimizing the model's hyperparameters in a technique called define-by-run. Furthermore, many commonly used network structures are tested in the 1D CNN form (Inception, MobileNet, LeNet, AlexNet, VGG16, and ResNet50). The proposed system is an end-to-end model that accepts raw 12-lead ECG signals (with a duration of 10 s and a sampling rate of 500 Hz) and produces 42 ECG signals. Figure 2 shows an example of a 12-lead ECG from the PTB-XL dataset for a patient with atrial fibrillation. The next section will explain the datasets and how each network structure was tested to build the system.

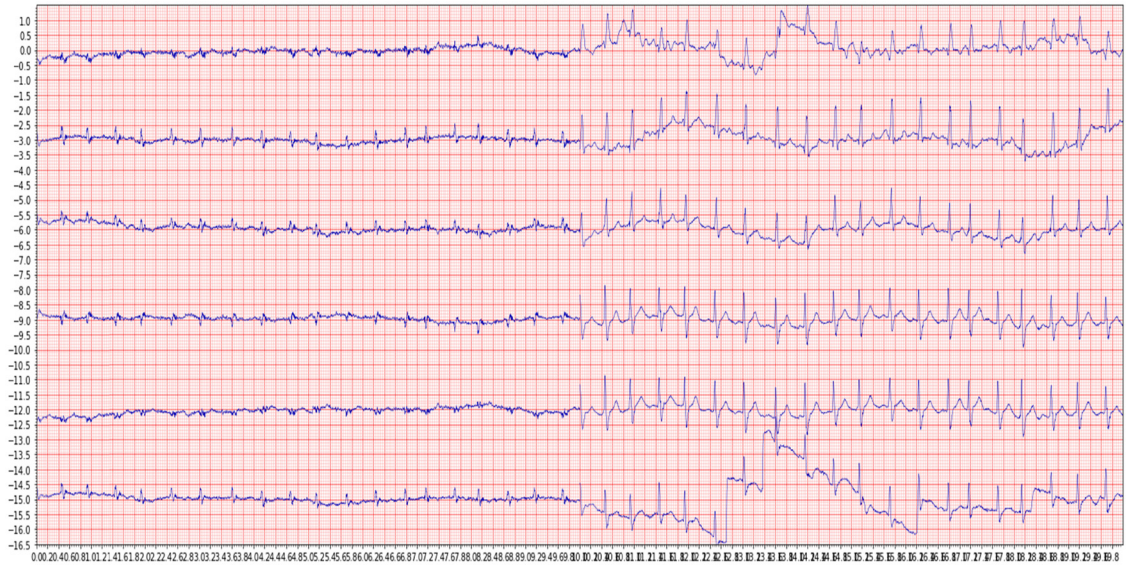

**Figure 2:** A sample of a 12-lead ECG from the PTB-XL dataset for a patient with atrial fibrillation [8].

## 3.1 Datasets

Three 12-lead datasets were used in this experiment to test the proposed system's generalization ability. The datasets come from different sources, each containing a different number of labels and records, as explained in Table 1. Furthermore, they are strongly imbalanced, making it a challenging classification task. Finally, we split the datasets for training and testing to test the models' performance on the test data they had not trained on before to avoid the overfitting problem.

## 3.2 Data preprocessing

The 12-lead ECG samples in the datasets come in different duration lengths, with a mean of 10 s. Since deep neural networks require the inputs to be of the same length, the datasets were preprocessed; thus, all samples were of 10 s length made by cropping the samples longer than 10–0 s and padding the samples less than 10 s with zeros.

## 3.3 Define-by-run

In the define-and-run technique, many hyperparameters must be defined before training a DL model [23]. On the other hand, in the define-by-run technique, the model's hyperparameters need not be defined before training the model. Instead, the DL model is implemented dynamically, and the hyperparameters are optimized in the context of an objective function with hyperparameters as inputs and the validation score as output [24]. The optimization algorithm used in this study is the tree of Parzen estimator (TPE) [25]. The TPE is a sequence model-based global optimization method that estimates the loss function using a probabilistic model and makes informed assumptions about the number of repetitions. The TPE method outperforms random search and grid search strategies for optimizing multiple hyperparameters, particularly for DL-based models with more hyperparameters than machine learning models [26].

The Bayes rule is used in the TPE, where the probability model $P(y|x) = P(x|y)P(y)/P(x)$, and $P(x|y)$ and is broken down into $h(x)$ and $j(x)$, which is calculated by:

**Table 1:** Description of the datasets used in this experiment

| Dataset | #Training records | #Testing records | Sex ratio (male/female) | Mean age (years) | Mean duration (s) | Sample frequency (Hz) | #Classes |
|---|---|---|---|---|---|---|---|
| Ningbo Hospital ECG Database [6] | 27,924 | 6,981 | 55.9%/54.1% | 51 | 10 | 500 | 42 |
| PTB-XL [8] | 17,469 | 4,368 | 52%/48% | 60 | 10 | 500 | 23 |
| Georgia ECG Challenge. Emory University, Atlanta, Georgia, USA [7] | 8,275 | 2,069 | 54%/46% | 61 | 10 | 500 | 24 |

$$P(x|y) := \begin{cases} h(x), & y < y_{\text{threshold}} \\ j(x), & y \geq y_{\text{threshold}}, \end{cases} \tag{1}$$

where $h(x)$ represents the objective function value that is less than the threshold for one parameter distribution, and $j(x)$ represents that the objective function value that is larger than the threshold for another hyperparameter distribution.

To choose the hyperparameters based on the probabilistic model, the expected improvement metric $E_i$ is used, which is calculated as follows:

$$E_i(x) = \int_{-\infty}^{y_{\text{threshold}}} (y_{\text{threshold}} - y) \frac{P(x|y)P(y)}{P(x)} \mathrm{d}y. \tag{2}$$

The hyperparameter set $x$ is expected to be improved when $E_i$ is positive. Thus, the TPE extracts a hyperparameter sample from $h(x)$, then makes an evaluation based on $h(x)/j(x)$, and finally, TPE returns the $x$ set that makes the best $E_i$ value.

#### 3.3.1 Optimized hyperparameters

Several hyperparameters are optimized to build the best model in our experiment:

1. The number of hidden CNN layers.
2. The number of kernels for each hidden layer.
3. The kernel size.
4. The number of dense layers.
5. The number of units in each dense layer.

   For all the previously mentioned hyperparameters, increasing the number increases the training time. Therefore, it may cause the model to overfit the training data, while decreasing it to a low number causes the model to underfit, which means that the model could not learn from the training data.
6. Activation function for each layer.
7. Batch size: the model's parameters are updated for each batch size of the training dataset [27]. The training time is affected significantly by changing the batch size. By choosing a large batch size, the training time may be reduced. On the other hand, increasing the batch size improves the convergence stability [28]. Thus, the batch size is a considerable parameter to be optimized.
8. Learning rate is a vital hyperparameter to determine the model's convergence to the global minimum. A higher learning rate value produces a worse loss value and may cause skipping of the global minimum point. However, a smaller learning rate value increases the training time and may cause the model to fall into a local minimum.
9. Optimizer: after calculating the loss value by the loss function, the optimizer minimizes the loss value.

#### 3.3.2 The proposed define-by-run model

Figure 3 depicts the entire workflow of the define-by-run model presented. The modeling process contains the following steps:

Step 1. The ECG dataset is used as input for the DL model.

Step 2. The TPE algorithm is applied to optimize the hyperparameters of the model.

Step 3. The model with the best hyperparameter combination is obtained, and the classification performance of the model is tested on the test set.

### 3.4 1D deep neural network models

In addition to the define-by-run proposed model, six widely used deep neural network models are tested in this study. These models are Inception [29], Mobilenet [30], LeNet, AlexNet [31], VGG16 [28], and ResNet50

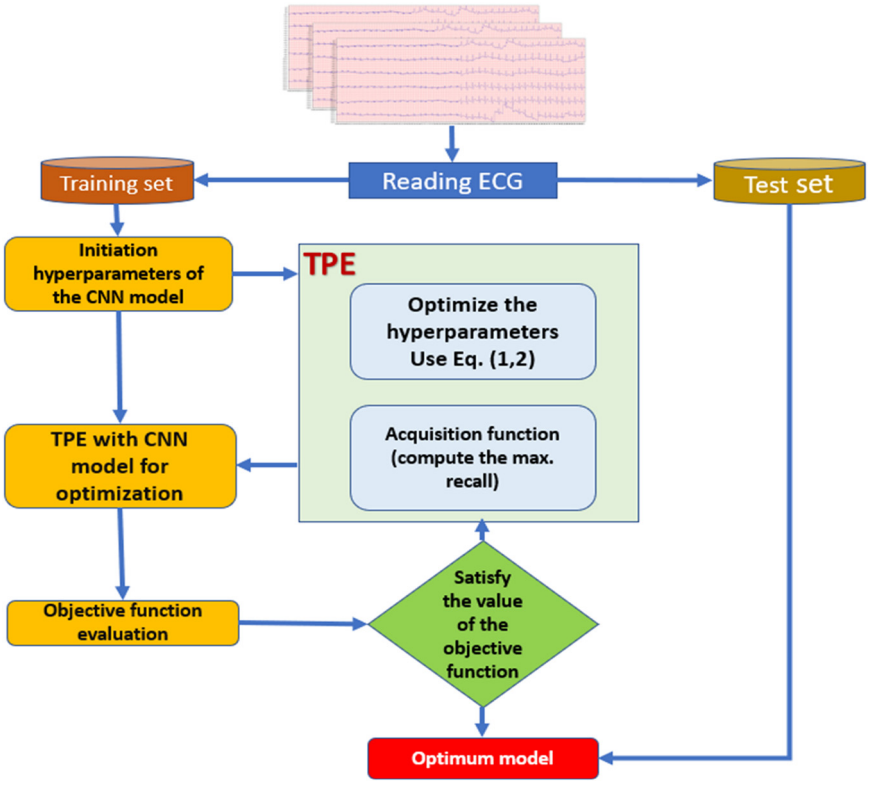


**Figure 3:** The workflow of the define-by-run model [Original].

[32]. Motivated by the success of these models in image classification, we constructed the same models using 1D CNN instead of 2D CNN. The input of each model is the ECG signal (with a dimension of $500 \times 12$) and the last layer (the Sigmoid activation function) with the number of classes of each dataset to make a multilabel classification. The models vary in layer construction, which makes the number of trainable parameters of each model different, as shown in Table 2, which affects the training time.

### 3.5 Loss function

This work used the binary cross-entropy as a loss function. Since most samples have more than one class at the same time, it figures out the probability of each class in each sample. Equation (3) shows how to determine the loss of binary cross-entropy:

$$L_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{Q}(\mathrm{pr}(x_{ij})\cdot\log l(x_{ij}) + (1 - \mathrm{pr}(x_{ij})\cdot\log(1 - l(x_{ij}))), \tag{3}$$

where $\mathrm{pr}(x)$ is the probability of class $x$ in the target, the probability of class $x$ in the prediction is represented by $l(x)$, the number of samples is indicated by $N$, and $Q$ is the number of classes.

## 4 Results and discussion

In this work, we used all proposed models randomly, and we employed Adam as an optimization function and 0.001 as an initial learning rate. If the validation data loss does not improve after one epoch, a lower learning rate with one patient is used for callbacks in order to decrease the learning rate by a factor of 0.5. In

**Table 2:** Number of the trainable parameters of each tested model

| Dataset | Model | No. of trainable parameters |
|---|---|---|
| Ningbo Hospital | Inception | 267,818 |
| | Mobilenet | 1,326,042 |
| | LeNet | 1,282,244 |
| | AlexNet | 179,608,522 |
| | VGG16 | 350,079,658 |
| | ResNet50 | 22,781,034 |
| | Define-by-run | 931,051 |
| PTB-XL | Inception | 266,583 |
| | Mobilenet | 1,301,703 |
| | LeNet | 1,281,617 |
| | AlexNet | 179,530,679 |
| | VGG16 | 350,001,815 |
| | ResNet50 | 19,706,967 |
| | Define-by-run | 3,241,667 |
| Georgia ECG | Inception | 266,648 |
| | Mobilenet | 1,302,984 |
| | LeNet | 1,281,650 |
| | AlexNet | 179,534,776 |
| | VGG16 | 350,005,912 |
| | ResNet50 | 19,868,760 |
| | Define-by-run | 3,241,808 |

addition, in order to minimize overfitting, the early stopping strategy with patient 2 is used if the validation loss of data has not improved after two epochs. The amount of time required to complete one cycle of training is shown in Table 3, which compares all of the models presented. The experiments were conducted on a 1.7 GHz Intel Core i7 processor with 16 GB RAM, 64-bit Windows 11 Professional, and an NVIDIA GeForce 4 GB display card. The programming language Python is employed with the TensorFlow library for building the models and the Optuna library for building the optimization algorithm.

**Table 3:** Training time in seconds of one epoch for each model

| Dataset | Model | Training time of one epoch (s) |
|---|---|---|
| Ningbo Hospital | Inception | 914 |
| | Mobilenet | 137 |
| | LeNet | 9 |
| | AlexNet | 228 |
| | VGG16 | 659 |
| | ResNet50 | 766 |
| | Define-by-run | 7 |
| PTB-XL | Inception | 413 |
| | Mobilenet | 78 |
| | LeNet | 7 |
| | AlexNet | 165 |
| | VGG16 | 552 |
| | ResNet50 | 881 |
| | Define-by-run | 50 |
| Georgia ECG | Inception | 193 |
| | Mobilenet | 28 |
| | LeNet | 3 |
| | AlexNet | 65 |
| | VGG16 | 193 |
| | ResNet50 | 199 |
| | Define-by-run | 10 |

## 4.1 Evaluation metrics

This study used four performance metrics that are commonly applied to evaluate the effectiveness of the proposed methods: accuracy ($\mathrm{Acc}_{\mathrm{multilabel}}$) (equation (4)), macro averaging recall ($R_{\mathrm{macro}}$) (equation (5)), macro averaging precision ($P_{\mathrm{macro}}$) (equation (6)), and macro averaging area under the union ($\mathrm{AUC}_{\mathrm{macro}}$):

$$\mathrm{Acc}_{\mathrm{multilabel}} = \frac{\mathrm{True}_{\mathrm{p}} + \mathrm{True}_{\mathrm{n}}}{\mathrm{True}_{\mathrm{p}} + \mathrm{False}_{\mathrm{p}} + \mathrm{True}_{\mathrm{n}} + \mathrm{False}_{\mathrm{n}}}, \tag{4}$$

$$R_{\mathrm{macro}} = \frac{\mathrm{True}_{\mathrm{p}}}{\mathrm{True}_{\mathrm{p}} + \mathrm{False}_{\mathrm{n}}}, \tag{5}$$

$$P_{\mathrm{macro}} = \frac{\mathrm{True}_{\mathrm{p}}}{\mathrm{True}_{\mathrm{p}} + \mathrm{False}_{\mathrm{p}}}, \tag{6}$$

where $\mathrm{True}_{\mathrm{p}}$, $\mathrm{True}_{\mathrm{n}}$, $\mathrm{False}_{\mathrm{p}}$, and $\mathrm{False}_{\mathrm{n}}$ denote the true positive, true negative, false positive, and false negative, respectively.

The average statistic that is aggregated across all of the available thresholds for classification is called the AUC. The value of the AUC ranges from 0 to 1, with higher values indicating the better overall performance of the model.

## 4.2 Define-by-run model

In this article, the TPE algorithm is used to optimize the hyperparameters of the CNN model using the define-by-run technique. The hyperparameters that enabled the model to obtain the highest performance were estimated depending on the optimization results. Table 4 shows the search space for hyperparameters

**Table 4:** Hyperparameter's search space and the selection of their adjustment of the CNN model

| Hyperparameter | Lower limit | Upper limit | Optimum value | Dataset |
|---|---|---|---|---|
| No. CNN layers | 1 | 4 | 3 | Ningbo Hospital |
| No. CNN kernels | [16, 32, 64, 128] | | 16 (all layers) | |
| CNN kernel size | [3, 5, 7] | | [5, 7, 3] (each layer, accordingly) | |
| No. dense layers | 1 | 2 | 2 | |
| No. dense layer units | 128 | 256 | [232, 210] (each layer, accordingly) | |
| Batch size | [128, 256] | | 128 | |
| Learning rate | 0.001 | 0.0001 | 0.0011 | |
| Activation function | [Relu, Selu, Gelu] | | [Relu, Selu, Gelu] (each layer, accordingly) | |
| Optimizer | [Adam, RMSprop] | | RMSprop | |
| No. CNN layers | 1 | 4 | 4 | PTB-XL |
| No. CNN kernels | [16, 32, 64, 128] | | [32, 128, 64, 64] (each layer, accordingly) | |
| CNN kernel size | [3, 5, 7] | | [7, 5, 7, 5] (each layer, accordingly) | |
| No. dense layers | 1 | 2 | 2 | |
| No. dense layer units | 128 | 256 | [156, 140] (each layer, accordingly) | |
| Batch size | [128, 256] | | 128 | |
| Learning rate | 0.001 | 0.0001 | 0.0092 | |
| Activation function | [Relu, Selu, Gelu] | | [Gelu, Selu, Selu, Selu] (each layer, accordingly) | |
| Optimizer | [Adam, RMSprop] | | Adam | |
| No. CNN layers | 1 | 4 | 4 | Georgia ECG |
| No. CNN kernels | [16, 32, 64, 128] | | [32, 128, 64, 64] (each layer, accordingly) | |
| CNN kernel size | [3, 5, 7] | | [7, 5, 7, 5] (each layer, accordingly) | |
| No. dense layers | 1 | 2 | 2 | |
| No. dense layer units | 128 | 256 | [156, 140] (each layer, accordingly) | |
| Batch size | [128, 256] | | 128 | |
| Learning rate | 0.001 | 0.0001 | 0.0092 | |
| Activation function | [Relu, Selu, Gelu] | | [Gelu, Selu, Selu, Selu] (each layer, accordingly) | |
| Optimizer | [Adam, RMSprop] | | Adam | |

and the best hyperparameters obtained for each dataset. By randomly initializing the hyperparameters, the TPE iterated 30 times while evaluating the model's recall with each iteration. Therefore, the optimal CNN model hyperparameters can be determined when the classifier has the highest recall. The recall iteration convergence plot for each dataset used in this study is shown in Figure 4.

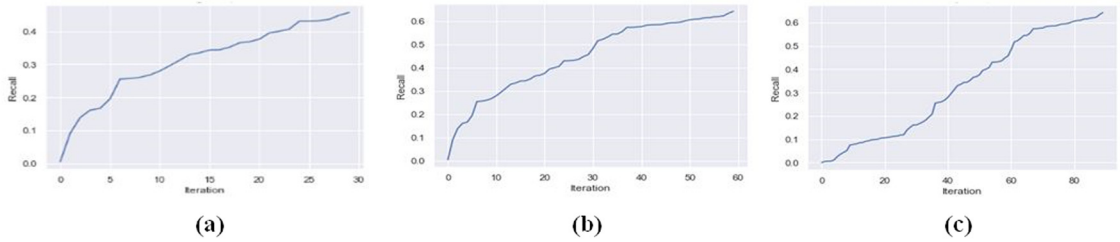


**Figure 4:** The recall iteration convergence plot for each dataset: (a) the Ningbo Hospital, (b) the PTB-XL, and (c) the Georgia ECG [Original].

Given that both datasets contain a similar number of classes, it is notable that the optimum hyperparameters for the PTB-XL and Georgia ECG datasets are the same (23 and 24, respectively). However, for the Ningbo Hospital dataset, the optimum hyperparameters are different since it has 42 classes.

Table 5 depicts the summary of the best model structure for the PTB-XL and Georgia ECG datasets. However, Table 6 depicts the best model structure summary for the Ningbo Hospital dataset.

**Table 5:** Summary of the best model structure for the PTB-XL and Georgia ECG datasets

| Layer type | Output shape |
|---|---|
| Input layer | (None, 5,000, 12) |
| Conv1D | (None, 5,000, 32) |
| Conv1D | (None, 2,500, 128) |
| Conv1D | (None, 1,250, 64) |
| Conv1D | (None, 625, 64) |
| Max pooling 1D | (None, 312, 64) |
| Flatten | (None, 19,968) |
| Dense | (None, 156) |
| Dense | (None, 140) |
| Dense (Sigmoid) | (None, 23) for PTB-XL, (None, 24) for Georgia ECG |

**Table 6:** Summary of the best model structure for the Ningbo Hospital dataset

| Layer type | Output shape |
|---|---|
| Input layer | (None, 5,000, 12) |
| Conv1D | (None, 2,500, 16) |
| Conv1D | (None, 2,500, 16) |
| Conv1D | (None, 2,500, 3) |
| Max pooling 1D | (None, 1,250, 3) |
| Flatten | (None, 3,750) |
| Dense | (None, 232) |
| Dense | (None, 210) |
| Dense (Sigmoid) | (None, 42) |

The performance metrics of the optimum model proposed by the define-by-run technique are illustrated in Table 7 with the number of training epochs and training time per epoch (all performance metrics were evaluated on test data). The results show that the best performance was achieved using the Ningbo dataset. In contrast, the weakest result was achieved using the Georgia ECG dataset since it has the lowest number of samples and some classes have just a few samples, which is insufficient to train the model and affects the recall ratio.

**Table 7:** Performance metrics of the optimum model proposed by the define-by-run technique

| Dataset | Accuracy (%) | Recall (%) | Precision (%) | AUC (%) | No. epochs | Time per epoch (s) |
|---|---|---|---|---|---|---|
| Ningbo Hospital | 97.33 | 46.81 | 97.71 | 68.52 | 27 | 7 |
| PTB-XL | 96.60 | 67.78 | 83.66 | 82.81 | 30 | 49 |
| Georgia ECG | 94.32 | 19.09 | 66.97 | 71.26 | 9 | 10 |

The training plots for validation and training data are shown in Figure 5. The plots show that the proposed model is more robust when trained on the Ningbo Hospital and the PTB-XL datasets. At the same time, the training process suffered some instability when the Georgia ECG dataset was used. Furthermore, the loss and accuracy plots show that the model started to converge after the first epoch.

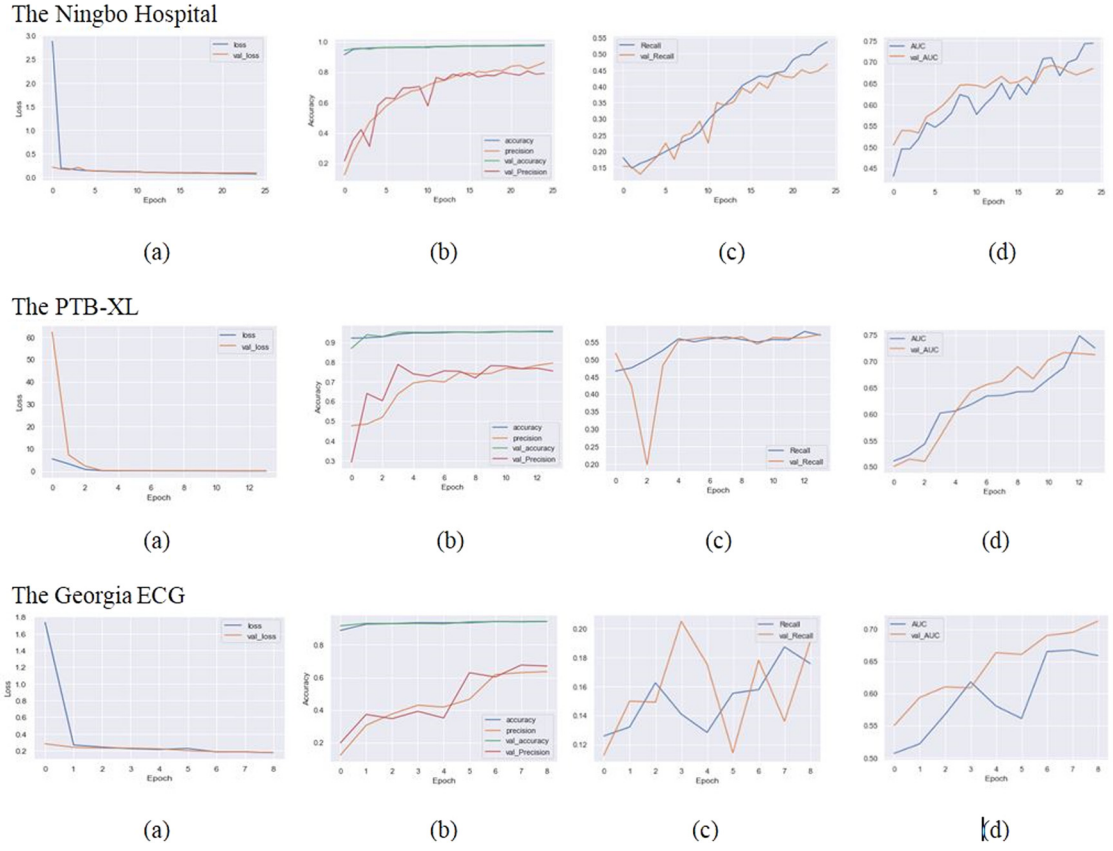


**Figure 5:** The training plots for validation and training data for the optimum models: (a) the loss plot, (b) the accuracy and precision plot, (c) the recall plot, and (d) the AUC plot [Original].

## 4.3 Deep neural network model results

The performance metrics for the tested deep neural network are shown in Table 8.

**Table 8:** Performance metrics for the tested deep neural network

| Dataset | Model | Accuracy (%) | Recall (%) | Precision (%) | AUC (%) | No. epochs | Time per epoch (s) |
|---|---|---|---|---|---|---|---|
| Ningbo Hospital | Inception | 97.89 | 53.69 | 90.83 | 74.32 | 11 | 916 |
| | Mobilenet | 97.90 | 56.22 | 87.75 | 77.75 | 21 | 138 |
| | LeNet | 97.55 | 49.92 | 83.38 | 72.74 | 14 | 9 |
| | AlexNet | 97.71 | 52.12 | 86.31 | 76.98 | 12 | 227 |
| | VGG16 | 96.73 | 25.59 | 81.68 | 68.05 | 13 | 651 |
| | ResNet50 | 95.89 | 23.81 | 49.27 | 66.20 | 14 | 751 |
| PTB-XL | Inception | 95.87 | 52.32 | 87.39 | 75.18 | 6 | 413 |
| | Mobilenet | 95.90 | 55.15 | 84.72 | 78.04 | 11 | 75 |
| | LeNet | 95.95 | 59.37 | 81.49 | 73.76 | 11 | 7 |
| | AlexNet | 96.53 | 64.39 | 85.67 | 84.49 | 18 | 166 |
| | VGG16 | 96.37 | 62.28 | 85.15 | 82.28 | 20 | 559 |
| | ResNet50 | 95.41 | 57.24 | 75.5 | 71.27 | 14 | 862 |
| Georgia ECG | Inception | 94.81 | 24.09 | 78.58 | 75.64 | 18 | 192 |
| | Mobilenet | 94.92 | 31.23 | 72.22 | 73.09 | 17 | 28 |
| | LeNet | 94.04 | 11.92 | 66.67 | 66.23 | 7 | 3 |
| | AlexNet | 95.02 | 35.49 | 70.71 | 77.53 | 19 | 67 |
| | VGG16 | 94.77 | 28.19 | 71.14 | 75.53 | 30 | 193 |
| | ResNet50 | 93.39 | 1 | 13.57 | 48.04 | 4 | 198 |

To understand the behavior of each tested model, Figure 6 shows the performance metrics plot for each dataset on train and test data. The results show that the AlexNet and Mobilenet models achieved the best performance. In contrast, the ResNet model suffered from the overfitting problem, which can be concluded due to the big variance between the train and test data in the training plots. On the other hand, the LeNet model achieved promising high results and close to the best results with the lowest training time, making it suitable to implement in low computational power and edge devices.

By examining the training plots, we can infer whether the following characteristics indicate overfitting:

- Decreasing training and validation loss: The model improves as it is trained. However, it is important to note that the training and validation loss should decrease simultaneously. If the training loss drops substantially more quickly than the validation loss, this may suggest the overfitting of the model.
- Stabilized losses in training and validation: Once the training and validation loss has stabilized, it signifies that the model has converged and is not overfitting. However, if the training loss drops while the validation loss remains stable or begins to grow, this may suggest overfitting.
- High accuracy of both training and validation: If both training and validation accuracies are good, then the model generalizes effectively and is not overfitting. On the other hand, if the training accuracy is significantly greater than the validation accuracy, this might be a signal of overfitting.
- Minimal to no difference between training and validation loss: If there is little to no difference between the training and validation loss, this implies no overfitting of the model. However, if there is a large difference between the two, it might be a symptom of overfitting.
- Stabilized training and validation loss: Once the training and validation loss has stabilized, it indicates that the model has converged and is not overfitting. However, if the training loss decreases while the validation loss stabilizes or even starts to increase, it may indicate overfitting.
- High training and validation accuracy: If both training and validation accuracies are high, the model is generalizing well and not overfitting. On the other hand, if the training accuracy is much higher than the validation accuracy, it could be an indication of overfitting.

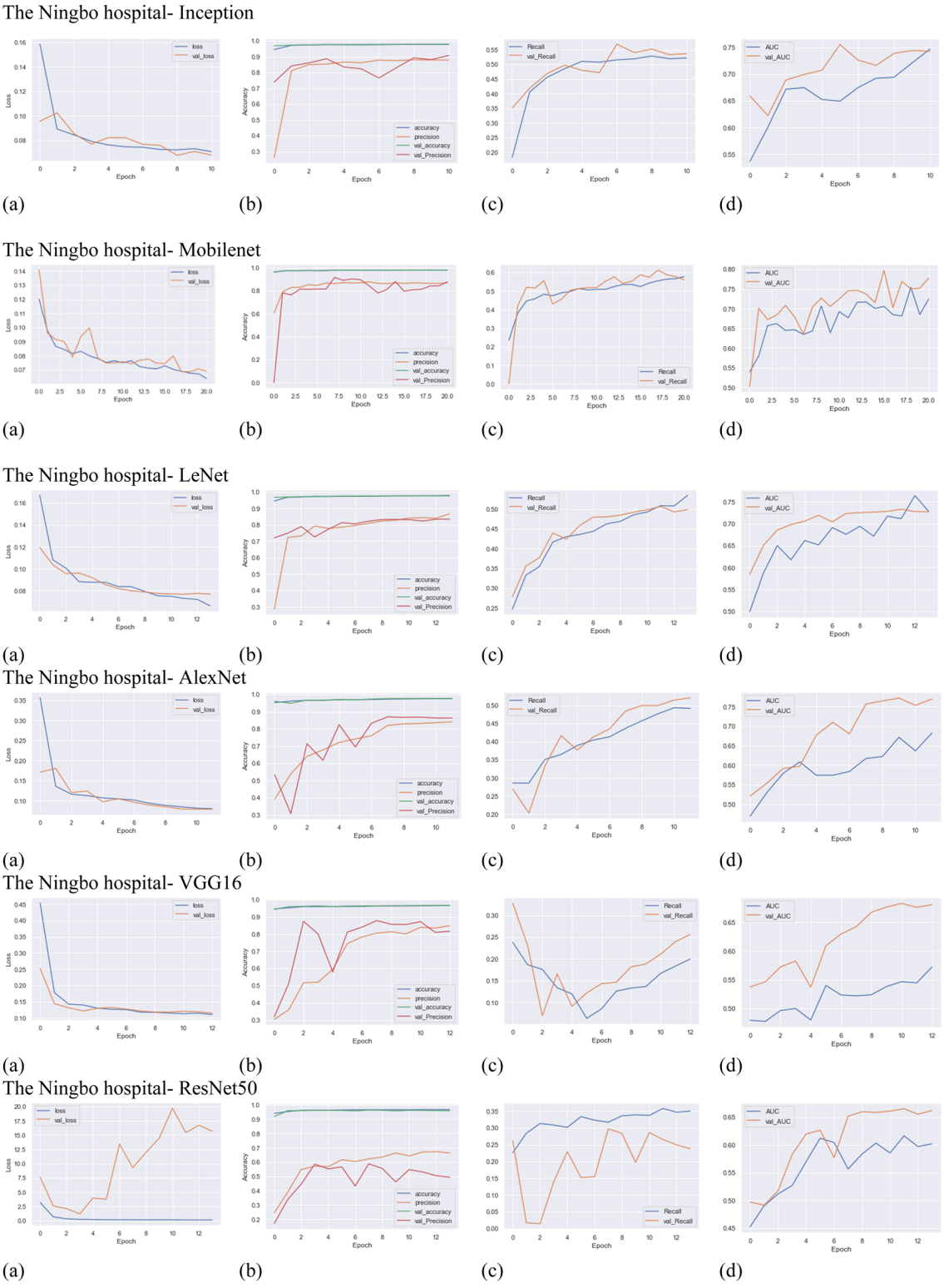


**Figure 6:** The training plot of the deep neural models on each dataset: (a) the loss plot, (b) the accuracy and precision plot, (c) the recall plot, and (d) the AUC plot [Original].

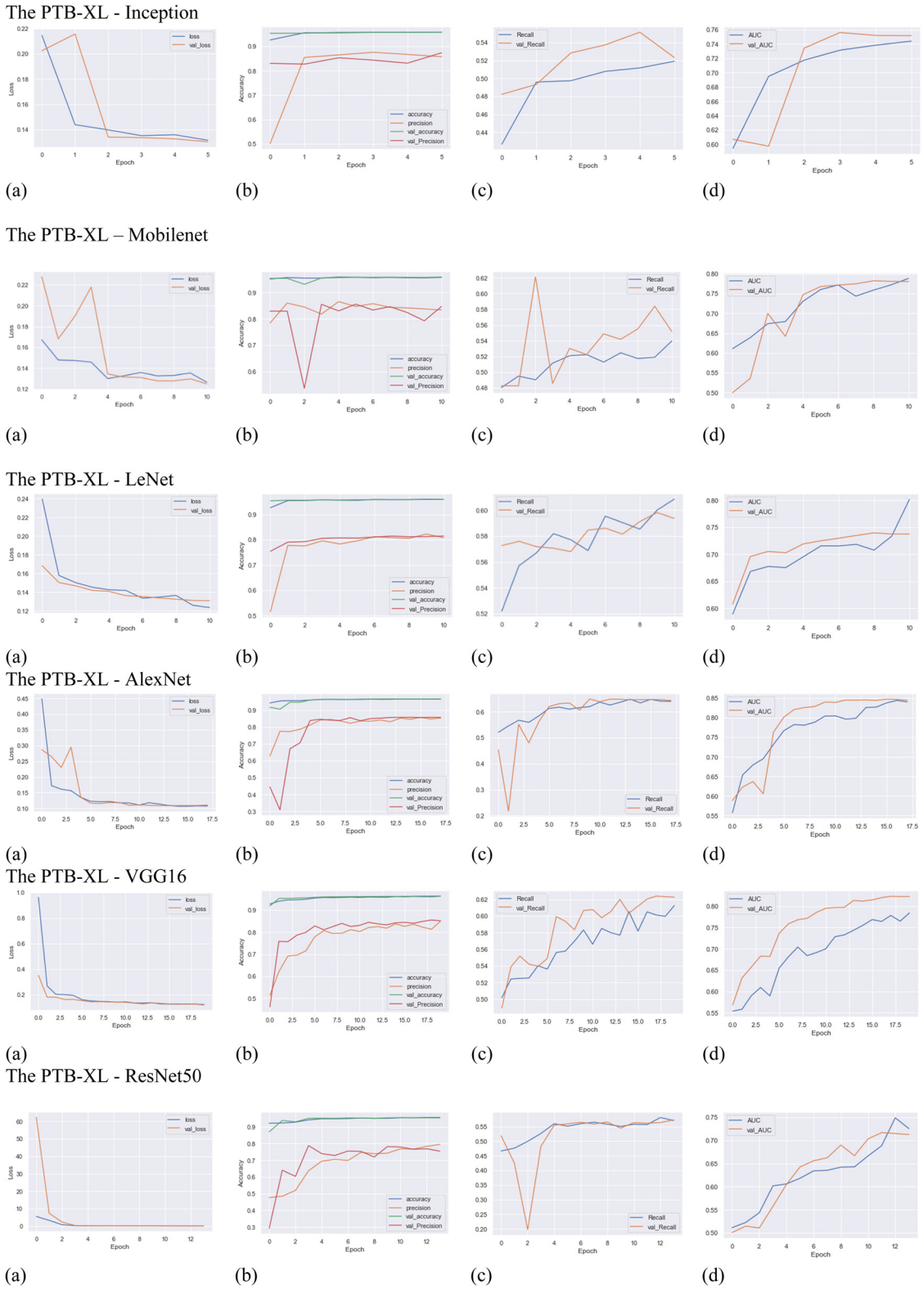


**Figure 6:** (*Continued*)

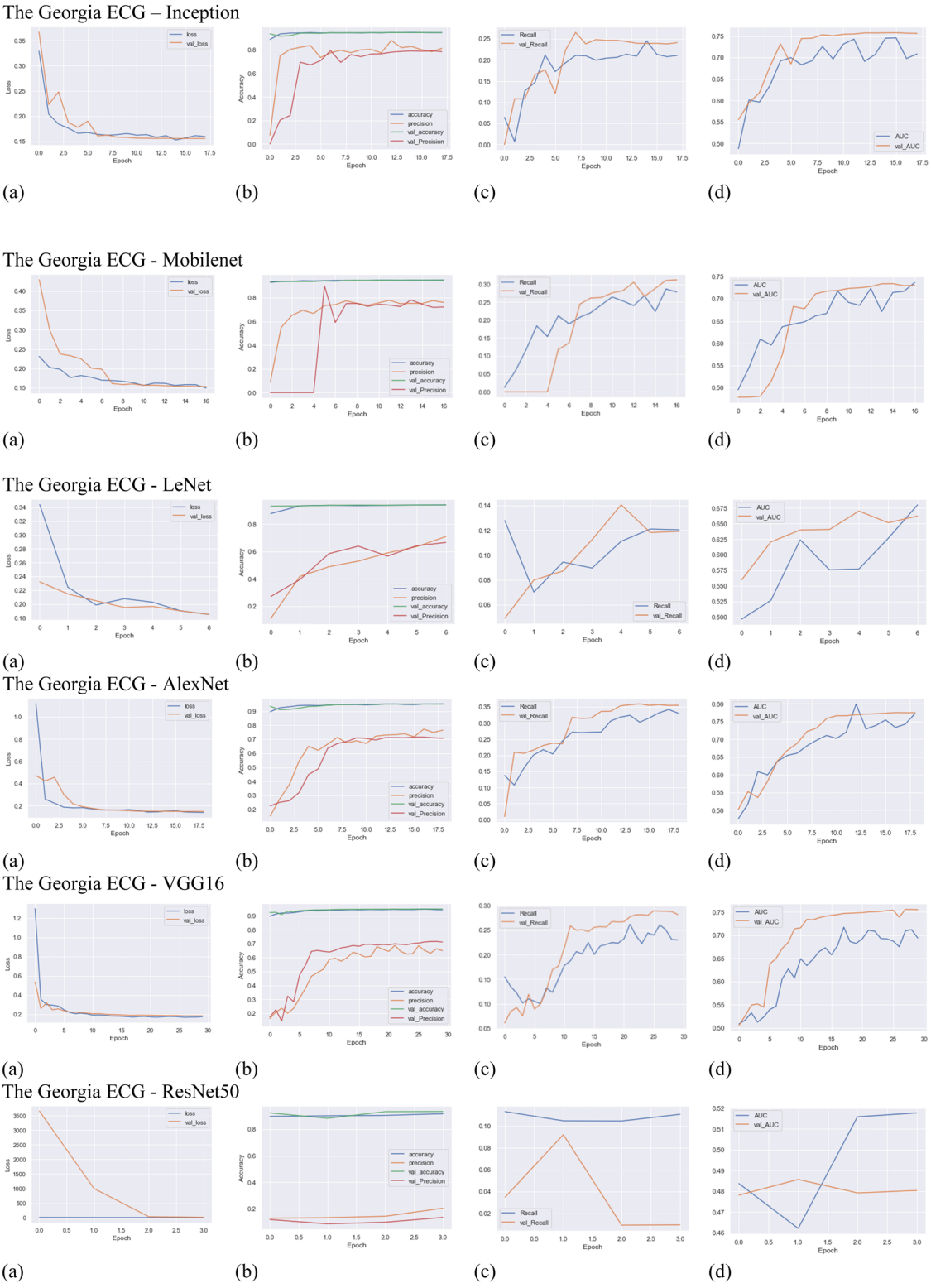


**Figure 6:** (*Continued*)

- Little or no gap between the training and validation loss: If there is little or no gap between the training and validation loss, it indicates that the model is not overfitting. However, if there is a significant gap between the two, it could be an indication of overfitting.

### 4.4 Comparison with the previous works

There are several approaches recommended in the literature for classifying ECG signals. However, the different numbers of classes and leads employed in each study should be considered while comparing the results. Table 9 shows the contrasts among the recommended techniques with some comparable strategies (N and MI stand for normal class and myocardial infarction, respectively). The comparison reveals that, despite classifying 42 multilabel classes and utilizing a bigger dataset, our technique topped all others in terms of performance.

**Table 9:** Comparison of the proposed method with some previous works

| Study | No. of leads | No. of classes | Dataset | Method | Accuracy (%) |
|---|---|---|---|---|---|
| [33] | 12 | 2 | N 31,722<br>MI 49,930 | CNN | 93.5 |
| [34] | 3 | 2 | N 5,000<br>MI 15,000 | Fourier/logistic regression | 95.6 |
| [35] | 12 | 3 | PTB and AF-challenge | CNN-LSTM | 94.6 |
| [36] | 12 | 7 | 7,704 samples | ResNet-LSTM | 81 |
| Define-by-run model | 12 | 42 | Ningbo | CNN | 97.33 |
| AlexNet | 12 | 42 | Ningbo | CNN | 97.90 |

## 5 Study limitations

Recall is a metric used in machine learning and statistics to quantify a model's capacity to identify all relevant instances (i.e., true positives) of a given class. The recall is the ratio of real positives to the total number of true positives, including both true and false negatives. In this study, there was no high recall. A poor recall indicates that the model fails to identify several relevant instances of the class in issue accurately. In other words, the model is omitting many actual positives or mistakenly identifying them as negatives (i.e., false negatives). In a medical diagnosis system, the recall may reflect the percentage of actual cases of illness that are correctly diagnosed. In this instance, a low recall rate would mean that the system fails to notice a significant number of occurrences, which might have disastrous consequences for the patient. The imbalance in the data is the primary contributor to the poor recall shown in this research. The recommended solution for improving poor recall due to imbalanced data is to gather additional data from a wider variety of sources for the classes that have a low sample size. This will allow the model to develop a more generalized and robust representation of the target class, leading to improved performance.

## 6 Conclusion

This study presents a 12-lead ECG classification method by building an end-to-end model. Theoretically, the study discussed building a model based on the CNN and the define-by-run technique and training it using three datasets from different sources to test the proposed model's generalization ability: the Ningbo Hospital, PTB-XL, and the Georgia ECG datasets. The accuracies achieved by the define-by-run model were 97.33, 96.60, and 94.32%, respectively. Furthermore, six deep neural network models were tested

(Inception, Mobilenet, LeNet, AlexNet, VGG16, and ResNet50), and the best results were achieved by Mobilenet and AlexNet models (with 97.9% accuracy for the Mobilenet model using the Ningbo dataset). Additionally, the comparison with the literature showed that our method outperformed the previous methods even though our method used multilabel classification and the previous methods handled the ECG classification as a multiclass classification problem. Since all the datasets used in the experiment were real-world data, we can practically assume that this technique may be developed and used in the medical industry in the real world or as a screening tool in situations or locations where access to a 12-lead ECG is unavailable. Furthermore, since we used a multilabel classification method (in contrast to most previous works), the proposed method can diagnose ECG records with more than one abnormality in the same ECG record, with a higher number of classes (42, 24, and 23). The proposed model is lightweight and can be implemented on limited-resource edge devices to build a real-time heart disease diagnosis system. However, we suggest solving the problem of the dataset imbalances in the dataset due to the existence of some rare heart abnormalities in future work. At the same time, this will solve the limitation of the low recall results in this study. Additionally, other optimization algorithms for hyperparameter tuning should be tested and compared to TPE optimization.

**Acknowledgments:** The authors would like to acknowledge the University of Gezira for its support. Also, the authors would like to thank all who supported us in completing this work.

**Funding information:** This research received no external funding.

**Author contributions:** Atiaf A. Rawi, Murtada K. Elbashir, and Awadallah M. Ahmed contributed to the design of the methodology. Atiaf A. Rawi developed the mathematical model and algorithms for the problem addressed in the manuscript. Atiaf A. Rawi, Murtada K. Elbashir, and Awadallah M. Ahmed collaborated on the simulation, result analysis, and interpretation of the findings. All authors jointly contributed to the writing and editing of the manuscript.

**Conflict of interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

**Ethical approval:** The manuscript does not report on or involve the use of any animal or human for this manuscript.

**Data availability statement:** The dataset used in this study taken from references.

# References

[1] Sarra RR, Dinar AM, Mohammed MA, Ghani MKA, Albahar MA. A robust framework for data generative and heart disease prediction based on efficient deep learning models. Diagnostics. 2022;12(12):2899.

[2] Sarra RR, Dinar AM, Mohammed MA, Abdulkareem KH. Enhanced heart disease prediction based on machine learning and χ2 statistical optimal feature selection model. Designs. 2022;6(5):87.

[3] Holst H, Ohlsson M, Peterson C, Edenbrandt L. A confident decision support system for interpreting electrocardiograms. Clin Physiol. 1999;19(5):410–8. doi: 10.1046/j.1365-2281.1999.00195.x.

[4] Bogun F, Anh D, Kalahasty G, Wissner E, Serhal CB, Bazzi R, et al. Misdiagnosis of atrial fibrillation and its clinical consequences. Am J Med. 2004;117(9):636–42. doi: 10.1016/j.amjmed.2004.06.024.

[5] WHO. Global status report on noncommunicable diseases. Geneva: World Health Organization; 2014. http://apps.who.int/medicinedocs/es/m/abstract/Js21756en/. 2014.

[6] Zheng J, Zhang J, Danioko S, Yao H, Guo H, Rakovski C. A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. Sci Data. 2020;7(1):48. doi: 10.1038/s41597-020-0386-x.

[7] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation. 2000;101(23):215–20. doi: 10.1161/01.cir.101.23.e215.

[8] Wagner P, Strodthoff N, Bousseljot RD, Kreiseler D, Lunze FI, Samek W, et al. PTB-XL, a large publicly available electrocardiography dataset. Sci Data. 2020;7(1):154. doi: 10.1038/s41597-020-0495-6.
[9] Murat F, Yildirim O, Talo M, Baloglu UB, Demir Y, Acharya UR. Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review. Comput Biol Med. 2020;120:103726. doi: 10.1016/j.compbiomed.2020.103726.
[10] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition; 2015. Published as a conference paper at ICLR 2015, p. 1409.1556.pdf (arxiv.org).
[11] Kang Y, Cai Z, Tan CW, Huang Q, Liu H. Natural language processing (NLP) in management research: A literature review. J Manag Analytics. 2020;7(2):139–72. doi: 10.1080/23270012.2020.1756939.
[12] Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, et al. A guide to deep learning in healthcare. Nat Med. 2019;25(1):24–9. doi: 10.1038/s41591-018-0316-z.
[13] Hong S, Zhou Y, Shang J, Xiao C, Sun J. Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review. Comput Biol Med. 2020;122:103801. doi: 10.1016/j.compbiomed.2020.103801.
[14] Chen TM, Huang CH, Shih ESC, Hu YF, Hwang MJ. Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model. iScience. 2020;23(3):100886. doi: 10.1016/j.isci.2020.100886.
[15] Datta S, Puri C, Mukherjee A, Banerjee R, Dutta Choudhury A, Singh R, et al. Identifying normal, AF and other abnormal ECG rhythms using a cascaded binary classifier. Comput Cardiol. 2017;44:1–4. doi: 10.22489/CinC.2017.173-154.
[16] Hannun AY, Rajpurkar P, Haghpanahi M, Tison GH, Bourn C, Turakhia MP, et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. Nat Med. 2019;25(1):65–9. doi: 10.1038/s41591-018-0268-3.
[17] He R, Liu Y, Wang K, Zhao N, Yuan Y, Li Q, et al. Automatic cardiac arrhythmia classification using combination of deep residual network and bidirectional LSTM. IEEE Access. 2019;7:102119–35. doi: 10.1109/ACCESS.2019.2931500.
[18] Strodthoff N, Wagner P, Schaeffer T, Samek W. Deep learning for ECG analysis: Benchmarks and insights from PTB-XL. IEEE J Biomed Health Inform. 2021;25(5):1519–28. doi: 10.1109/JBHI.2020.3022989.
[19] Ullah A, Anwar SM, Bilal M, Mehmood RM. Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation. Remote Sens. 2020;12(10):1685. doi: 10.3390/rs12101685.
[20] Zhang D, Yang S, Yuan X, Zhang P. Interpretable deep learning for automatic diagnosis of 12-lead electrocardiogram. iScience. 2021;24(4):102373. doi: 10.1016/j.isci.2021.102373.
[21] He Z, Yuan Z, An P, Zhao J, Du B. MFB-LANN: A light-weight and updatable myocardial infarction diagnosis system based on convolutional neural networks and active learning. Comput Methods Prog Biomed. 2021;210:106379. doi: 10.1016/j.cmpb.2021.106379.
[22] Pan J, Tompkins WJ. A real-time QRS detection algorithm. IEEE Trans Biomed Eng. 1985;BME-32(3):230–6. doi: 10.1109/TBME.1985.325532.
[23] Lago J, de Ridder F, de Schutter B. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. Appl Energy. 2018;221:386–405. doi: 10.1016/j.apenergy.2018.02.069.
[24] Zhang J, Meng Y, Wei J, Chen J, Qin J. A novel hybrid deep learning model for sugar price forecasting based on time series decomposition. Math Probl Eng. 2021;2021:1–9. doi: 10.1155/2021/6507688.
[25] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. NIPS'11: Proceedings of the 24th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2011; p. 2546–2554.
[26] Swanson K, Trivedi S, Lequieu J, Swanson K, Kondor R. Deep learning for automated classification and characterization of amorphous materials. Soft Matter. 2020;16(2):435–46. doi: 10.1039/c9sm01903k.
[27] Nakama T. Theoretical analysis of batch and on-line training for gradient descent learning in neural networks. Neurocomputing. 2009;73:1–3. doi: 10.1016/j.neucom.2009.05.017.
[28] Parikh N. Accurate, Large Minibatch SGD: Training imagenet in 1 hour (FIXME). Found Trends® Optim. 2014;1(3):127–239.
[29] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-ResNet and the impact of residual connections on learning. 31st AAAI Conference on Artificial Intelligence, AAAI 2017; 2017. p. 4278–84.
[30] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted residuals and linear bottlenecks. USA: Computer vision and pattern recognition; 2018. p. 4510–20. doi: 10.1109/CVPR.2018.00474.
[31] Jiang X, Hu B, Chandra Satapathy S, Wang SH, Zhang YD. Fingerspelling Identification for Chinese Sign Language via AlexNet-Based Transfer Learning and Adam Optimizer. Sci Program. 2020;2020:1–13. doi: 10.1155/2020/3291426.
[32] Abedalla A, Abdullah M, Al-Ayyoub M, Benkhelifa E. Chest X-ray pneumothorax segmentation using U-Net with EfficientNet and ResNet architectures. PeerJ Comput Sci. 2021;7:607. doi: 10.7717/peerj-cs.607.
[33] Lodhi AM, Qureshi AN, Sharif U, Ashiq Z. A novel approach using voting from ECG leads to detect myocardial infarction. Adv Intell Syst Comput. 2018;869:337–52. doi: 10.1007/978-3-030-01057-7_27.
[34] Sadhukhan D, Pal S, Mitra M. Automated identification of myocardial infarction using harmonic phase distribution pattern of ECG Data. IEEE Trans Instrum Meas. 2018;67(10):2303–13. doi: 10.1109/TIM.2018.2816458.
[35] Lui HW, Chow KL. Multi-class classification of myocardial infarction with convolutional and recurrent neural networks for portable ECG devices. Inform Med Unlocked. 2018;13:26–33. doi: 10.1016/j.imu.2018.08.002.
[36] Chen YJ, Liu CL, Tseng VS, Hu YF, Chen SA. Large-scale classification of 12-lead ECG with deep learning. USA: IEEE EMBS International Conference on Biomedical & Health Informatics; 2019. p. 1–4. doi: 10.1109/BHI.2019.8834468.