Research Article

Azhar F. Al-zubidi*, Alaa Kadhim Farhan, and Sayed M. Towfek

# Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model

https://doi.org/10.1515/jisys-2023-0195
received October 15, 2023; accepted December 23, 2023

**Abstract:** Network security faces increasing threats from denial of service (DoS) and distributed denial of service (DDoS) attacks. The current solutions have not been able to predict and mitigate these threats with enough accuracy. A novel and effective solution for predicting DoS and DDoS attacks in network security scenarios is presented in this work by employing an effective model, called CNN-LSTM-XGBoost, which is an innovative hybrid approach designed for intrusion detection in network security. The system is applied and analyzed to three datasets: CICIDS-001, CIC-IDS2017, and CIC-IDS2018. We preprocess the data by removing null and duplicate data, handling imbalanced data, and selecting the most relevant features using correlation-based feature selection. The system is evaluated using accuracy, precision, $F1$ score, and recall. The system achieves a higher accuracy of 98.3% for CICIDS-001, 99.2% for CICIDS2017, and 99.3% for CIC-ID2018, compared to other existing algorithms. The system also reduces the overfitting of the model using the most important features. This study shows that the proposed system is an effective and efficient solution for network attack detection and classification.

**Keywords:** cyberattack prediction, cyberattack classification, CNN-LSTM-XGBoost model, correlation-based feature selection

## 1 Introduction

Cybersecurity is one of the important challenges in computer networks, as networks are vulnerable to various types of attacks that can compromise their functionality and integrity. Intrusion detection systems (IDSs) are intended to track network traffic and find any anomalous or malicious activity. However, traditional IDSs have some drawbacks, such as low prediction accuracy and inability to cope with new or unknown attacks. To overcome these problems, IDSs gained advantages from the application of machine learning (ML) and deep learning (DL) strategies. These advanced approaches are useful for dynamic threat environments because they learn from data instead of depending on defined rules or signatures. We review the most recent IDSs with ML- and DL-based in this literature review, rather than analyzing their efficacy and performance in detecting different types of network intrusions [1,2].

*** Corresponding author: Azhar F. Al-zubidi,** Computer Science Department, College of Sciences, AL Nahrain University, Jadriya, Baghdad, Iraq; Computer Sciences Department, University of Technology, Baghdad 10066, Iraq,
e-mail: cs.21.07@grad.uotechnology.edu.iq
**Alaa Kadhim Farhan:** Computer Sciences Department, University of Technology, Baghdad 10066, Iraq,
e-mail: alaa.k.farhan@uotechnology.edu.iq
**Sayed M. Towfek:** Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, 35511, Egypt, e-mail: profsm@nafsy.net

We intend in this review to provide an overview of the state-of-the-art IDSs and compare their performance and effectiveness in detecting different types of network attacks. Traditional IDSs include disadvantages such as poor prediction precision, high false-positive/negative rates, and an inability to protect to unexpected or unidentified attacks. We also identify some research gaps and challenges that need further investigation and propose some research objectives for this study [3]. The following are the criteria we used to choose the literature for this review: (1) the articles that published in peer-reviewed journals in the past 5 years; (2) they focused on ML- or DL-based IDSs for network security; (3) they used publicly available datasets that contain up-to-date and diverse attack types; and (4) they reported relevant performance metrics [4,5], such as accuracy, *F*1-score, precision, and recall [2]. The main contributions of this work are as follows:

1. The CICIDS-001, CICIDS2017, and CICIDS2018 datasets used in testing the IDSs were preferred.
2. Class imbalance was minimized by reducing the imbalance rates of the dataset in the preprocessing stage.
3. Anomaly detection was performed using DL algorithms. In the model, two hidden layers with 32 kernels each were created using convolutional neural network (CNN) and long short-term memory (LSTM) algorithms. The task of these kernels is to record the data learned during the training.
4. Using network traffic data, we used the XGBoost algorithm, which is renowned for its efficiency, speed, and low dependency on CPU resources, to identify anomalies and other threats.
5. At the end of the study, precision, *F*1 score, accuracy, and recall performances were measured on different datasets.

The structure of the remaining study is as follows:

Section 2 is the literature review that offers a thorough examination of ML-based IDSs; Section 3 is datasets and preprocessing, which is an overview of the datasets and preprocessing techniques that used. Section 4 is the methodology that describes design, strategies for implementation, and measures for the evaluation of our study. Section 5 is the hybrid proposed system. Section 6 is the results and discussion. And Section 7 is the conclusion.

## 2 Literature review

In this section, we reviewed multiple research that uses ML and DL techniques to find intrusions and anomalies in network traffic data. Different datasets, including UNSW-NB15, NSL-KDD, CICIDS-2017, and CIC-IDS-2018, as well as various techniques, including k-nearest neighbor (KNN), support vector machine (SVM), CNN, and LSTM, are used in the study. To assess their performance, the studies also provide other measures, such as accuracy, sensitivity, and *F*1-score.

In the study by Jasem and Jawhar [6], DL-based IDSs have been proposed to extract both spatial and temporal information from network data, for instance, researchers created a hybrid model that combines LSTM and CNN algorithms, and to evaluate their model using CIC-IDS2017, UNSW-NB15, and WSN-DS datasets, and achieved high accuracy and detection rates for binary and multiclass classification scenarios.

Bingu and Jothilakshmi [7] proposed a network intrusion detection systems (NIDS) based on CNN and transfer learning models to increase the performance and robustness of anomaly detection models for software-defined networking (SDN) environments. They tried and evaluated their model using the InSDN benchmark dataset and achieved high accuracy and detection rates. Similarly, to balance network traffic, a NIDS based on CNN and the synthetic minority oversampling technique-edited nearest neighbors algorithm, along with synthetic minority oversampling, was developed. They evaluated their model using the NSL-KDD dataset and had very good classification accuracy.

Thaseen et al. [8] introduced an ensemble-based DL technique that combined K-means with LSTM, CNN, recurrent neural network (RNN), gated recurrent unit, and deep neural network (DNN) classifiers to detect distributed denial of service (DDoS) attacks in CICIDS 2018 and SDN-based DDoS attack datasets. They used Random Forest for feature selection (FS) and achieved high accuracy and precision.

Manthiramoorthy and Khan [9] compared multiple encrypted cloud storage platforms and focused on the safety concerns of popular cloud storage systems for cryptography, such as Google Cloud, Microsoft Azure,

Tresorit, and Amazon S3. Privacy policies, key management, access control, and data encryption are the main topics of the analysis.

Ameen et al. [10] examined the blockchain (BC) and the dimensions of artificial intelligence (AI) techniques. The roles of BC, cybersecurity, and AI in the Internet of medical devices. They emphasized decentralized alternatives for safe healthcare applications while addressing opportunities, difficulties, and research goals.

Alshingiti et al. [11] displayed an ML-based network IDSs that can distinguish between malicious and normal traffic. Using NSL-KDD, the effectiveness of SVM, J48, Random Forest, and naive bytes was examined. They also claimed that Random Forest outperforms state-of-the-art IDSs.

Zivkovic et al. [12] presented a DL system that can detect phishing websites by analyzing their uniform resource locators (URLs). They used three algorithms: LSTM, CNN, and a hybrid of LSTM and CNN. They compared the performance of these algorithms on a dataset of phishing and legitimate URLs and indicated that CNN has the best accuracy rate, achieving 99.2%. They also discussed the challenges and future work of phishing detection using DL techniques.

Ozcan et al. [13] described a DL-based system that can identify chest X-ray images with coronavirus disease 2019 (COVID-19) infection. In this study, features from the photographs are extracted using a simple CNN model, and the images are then classified using an XGBoost classifier. The XGBoost hyperparameters are tuned in this study using a hybrid version of the arithmetic optimization method. They evaluated the performance of the proposed system on a balanced dataset of normal, X-ray images of COVID-19 and viral pneumonia, which indicates its high accuracy and precision.

Cai et al. [14] presented an ML-based system that can identify advanced persistent threat attacks on network traffic. To distinguish between normal and malicious network traffic, they used an XGBoost and Random Forest classifiers hybrid ensemble ML model. The effectiveness of the suggested system is evaluated using four online accessible datasets, namely, CSE-CIC-IDS2018, UNSW-NB15, CIC-IDS2017, and NSL-KDD. The study indicated that the hybrid ensemble model surpasses other recent research in the literature in terms of high accuracy and low false-positive rate for all datasets.

Sun et al. [15] presented a DL system that can predict the phishing URL by analyzing their characters and natural language features. They used hybrid DL models that combine DNN algorithms and LSTM to extract and classify features from the URL. The performance of the suggested models was evaluated in this study on phishing datasets and showed that they achieve high accuracy and outperform other phishing detection models. They also discussed how to combine different feature extraction techniques to improve phishing detection performance.

Sangodoyin et al. [16] used a DL framework that can predict malicious network traffic by analyzing their spatial and temporal features. They used a hybrid parallel DL model that includes two parallel LSTMs and two parallel CNNs to extract and classify features from network traffic. They also used a margin learning-based classifier (CosMargin) to increase the classification accuracy. They evaluated the performance of the proposed system on different malicious classes and showed that it achieves high detection accuracy and outperforms other models.

Oleiwi et al. [17] presented a DL system that can detect network intrusions by examining the spatial and temporal characteristics of network traffic data. They used a hybrid network of LSTM and CNNs to extract and classify features from network traffic data. They also used a category weight optimization technique to increase the resilience of the system and evaluated the performance of the system on CICIDS2017.

Manickam et al. [18] developed dataset that is used to test, benchmark, and fine-tune detection systems called internet control message protocol version 6 (ICMPv6)–DDoS attack dataset; they generated ICMPv6–DDoS attacks based on Router Advertisement and Neighbor Solicitation message flooding, in addition to normal and abnormal ICMPv6 traffic, including ten distinct, by utilizing a GNS3 network simulation tool. The suggested dataset has a minimal false-positive rate and high detection accuracy, effectively representing attack traffic. In general, this approach ensures that future researchers can still use the current IPv6 datasets by addressing their constraints.

The goal of the research by Alghuraibawi et al. [19] was on IPv6 network anomaly detection of DDoS assaults against ICMPv6 messages. Even with IPv6's security improvements, DDoS attacks and other vulnerabilities from IPv4 still present. In order to enhance the accuracy of ICMPv6–DDoS attack identification, the suggested method optimizes the subset of features through the use of bio-inspired algorithms for FS. The

effectiveness of this technique is demonstrated by the results, which address a key problem in network security and shield people and businesses from financial harm.

[20] FS for Binary Flower Pollination Algorithm (BFPA-FA)-Based ICMPv6-Based DDoS Attack Detection. Science & Engineering of Computer Systems. In this study, a novel technique based on the BFPA-FA is presented to identifying ICMPv6–DDoS flooding attacks. A SVM and FS together yield an impressive 97.96% accuracy rate in identifying ICMPv6–DDoS attacks. Interestingly, it shows efficiency by cutting the total characteristics from 19 to just 9. Using IDS to improve network security, this tried-and-true technique protects against dangerous attacks.

Nuiaa et al. [21] presented a novel technique for identifying Lightweight Directory Access Protocol (LDAP)-based DDoS attacks. By flooding open LDAP servers with increased traffic, these attacks take advantage of them. The suggested model employed an improved particle swarm optimization technique known as AWTPSO, which combines the properties of the LDAP protocol and network traffic features to detect attack patterns. Feature thresholds are dynamically adjusted by an adaptive weighted threshold model. Result shows an exceptional 99.99% detection accuracy with only 0.01% false positives, outperforming other state-of-the-art techniques. This effective solution successfully defends company networks against the threat posed by LDAP-based distributed reflection denial-of-service assaults.

Alghuraibawi et al. [22] offered a technique for identifying ICMPv6 attacks with DDoS on IPv6 networks. Even with its enhanced functionality, IPv6 is still susceptible to DoS and DDoS attacks. For FS, the method makes use of a modified Flower Pollination Algorithm (MFPA). The objective is to identify ICMPv6 DDoS assaults by extracting the most appropriate features from the ICMPv6 dataset. They also presented a multi-objective fire program analysis (FPA) model, which enhances the detection accuracy even more. The results obtained show an impressive 97.96% accuracy utilizing the MFPA with 10 features and 97.01% accuracy using the multi-objective FPA with only five features. This efficient solution extends network defenses against DDoS attacks using ICMPv6.

# 3 Datasets and preprocessing

In this work, we used three datasets CICIDS-001, CIC-IDS2017, and CIC-IDS2018, which are two modern labeled datasets for testing IDSs. The datasets contain normal and malicious network flows collected from an emulated network environment with 50 servers and various attack scenarios, such as DoS, DDoS, infiltration, and botnet. These datasets were preferred due to their widespread use in IDS research. Each dataset offered unique challenges, allowing us to thoroughly evaluate our hybrid model's performance across different scenarios. The datasets are divided into several days, each pertaining to a different attack class [23].

## 3.1 CICIDS-001 dataset

The CICIDS-001 dataset is a recent and realistic dataset that contains network traffic data for IDSs in a cloud environment. The dataset was created by Ring et al. in 2018 using the OpenStack platform to simulate a network with several clients and servers, such as email servers and web servers [24]. The primary objective of using this dataset was to assess the model's accuracy and its ability to handle diverse attack scenarios. The dataset captures both normal and malicious traffic, with a variety of attacks injected into the network by using predefined profiles or custom scripts. The dataset is divided into four parts, each corresponding to one week of data collection. The dataset is in a flow-based format, using the unidirectional NetFlow standard 2 to record the statistics of each network flow. The dataset contains 14 attributes, namely, 10 default NetFlow attributes and 4 additional attributes that provide more information about the flow direction, status, label, and sublabel. The dataset contains approximately 2.8 million records, with 79 types of attacks classified into four categories: port scan, ping scan, denial of service (DoS), and brute force (Table 1). It is publicly available online and can be applied for testing IDSs and evaluating it, especially for anomaly-based detection techniques [25].

**Table 1:** CICIDS-001 attack types

| Attack name | Number of instances |
|---|---|
| Normal | 2,096,134 |
| Attacker | 172,846 |
| Victim | 128,016 |
| Suspicious | 90,819 |
| Unknown | 10,286 |

## 3.2 CIC-IDS2017 dataset

This dataset was generated by the Canadian Institute for Cybersecurity (CIC) in 2017 to provide a realistic and up-to-date benchmark for NIDS. It consists of two parts: PCAP files and CSV files. The PCAP files contain the raw network packets captured from a simulated network environment, while the CSV files contain the extracted features and labels for each network flow. The dataset covers both normal and malicious traffic, with a total of eight attack scenarios: brute force, FTP, heartbleed, DoS, brute force SSH, web attack, infiltration, botnet, and DDoS (Table 2). It is generated by using a user profiling system called the B-Profile, which can mimic the abstract behavior of human interactions based on different protocols, such as HTTP, FTP and email. The dataset contains 16,545,685 records with 80 features for each record. The features include much information, such as timestamps, protocols, source, destination IP addresses, and derived information, such as packet length, duration, flags, payload size, and entropy. It was publicly available on CIC website and Kaggle. Our focus with this dataset was to address class imbalance and enhance precision. It can be used for various purposes, such as testing and evaluating NIDS techniques, conducting feature analysis and selection, applying ML algorithms, and performing data mining and visualization [26].

**Table 2:** CIC-IDS2017 attack types

| Attack name | Number of instances |
|---|---|
| BENIGN | 2,096,134 |
| DoS Hulk | 172,846 |
| DDoS | 128,016 |
| Portscan | 90,819 |
| DoS (GoldenEye) | 10,286 |
| FTP-Patator | 5,933 |
| DoS (Slowloris) | 5,385 |
| DoS (Slowhttptest) | 5,228 |
| SSH-Patator | 3,219 |
| Bot | 1,953 |
| WA(brute force) | 1,470 |
| WA(XSS) | 652 |
| Infiltration | 36 |
| WA (sql Injection) | 21 |
| Heartbleed | 11 |

## 3.3 CIC-IDS2018 dataset

This dataset was initialized by the Canadian Institute for Cybersecurity (CIC) in 2017 to provide a realistic and up-to-date benchmark for networks (NIDS). The dataset consists of two parts: PCAP files and CSV files. The PCAP files contain the raw network packets captured from a simulated network environment, while the CSV files contain the extracted features and labels for each network flow. The dataset covers both normal and malicious traffic, with a total of eight attack scenarios: heartbleed, (DDoS), brute force FTP, web attack,

infiltration, brute force SSH, DoS, and botnet (Table 3). The dataset was generated by using a user profiling system called B-Profile, which can mimic the abstract behavior of human interactions based on different protocols [13]. The dataset contains 16,545,685 records with 80 features for each record. Itb includes basic information such as source and addresses, destination IP, protocols, ports, and timestamps, as well as derived information such as packet length, duration, flags, payload size, and entropy. The dataset is publicly available on Kaggle. It can be used for various purposes, such as testing and evaluating NIDS techniques, conducting feature analysis and selection, applying ML algorithms, and performing data mining and visualization [27].

**Table 3:** CIC-IDS2018 attack types

| Attack name | Number of instances |
|---|---|
| BENIGN | 2,856,035 |
| DDoS (attack-HOIC) | 686,232 |
| DDoS (attack-LOIC-HTTP) | 1,152 |
| DoS (attack-Hulk) | 654,301 |
| Bot | 286,191 |
| FTP (brute force) | 387 |
| SSH (brute force) | 375 |
| Infiltration | 93,063 |
| DoS (attacks-SlowHTTPtest) | 280 |
| DoS (attacks-GoldenEye) | 83 |
| DoS (attacks-Slowloris) | 22 |
| DoS attacks-LOIC-UDP | 3 |
| Brute force-Web | 1 |

## 3.4 Comparison of the CICIDS-2018 and CICIDS-2017 datasets

The CICIDS-2018 dataset is a current and reliable dataset that contains various types of network intrusions and cyberattacks. It is an improved version of the CICIDS-2017 dataset, which was also produced by the Amazon Web Services platform. The CICIDS-2018 dataset has more data sources, more attack scenarios, and more extracted features than the CICIDS-2017 dataset. It also has a larger size and a higher diversity of network traffic. The CICIDS-2018 dataset consists of 80 statistical features that are computed in forward and backward directions, such as flow, packet length, and duration. The dataset has about five million records and is available online in PCAP and CSV formats. In this study, we used the CSV format [10]. The CICIDS-2018 dataset includes two patterns and seven attack techniques, such as brute force, heartbleed, DDoS attacks, DoS attacks, infiltration, botnet, brute force SSH, and web attacks [15]. The dataset architecture includes 50 systems, while the attacking entities include 31 servers. Table 4 shows the comparison between the two datasets. However, the number of web attacks in CICIDS-2018 is very low [28].

**Table 4:** CIC-IDS2017 and CIC-IDS2018 attack comparison

| Attack types | CIC-CIDS2017 | CIC-IDS2018 |
|---|---|---|
| Normal | 1,743,179 | 6,112,151 |
| DDoS | 128,027 | 687,742 |
| Brute force | 13,835 | 380,949 |
| Infiltration | 36 | 161,934 |
| DoS | 252,661 | 654,301 |
| Port scan | 158,930 | — |
| Web attacks | 2,180 | 928 |
| Botnet | 1,966 | 286,191 |

## 3.5 Preprocessing

Figure 1 shows multiple preprocessing steps, such as:

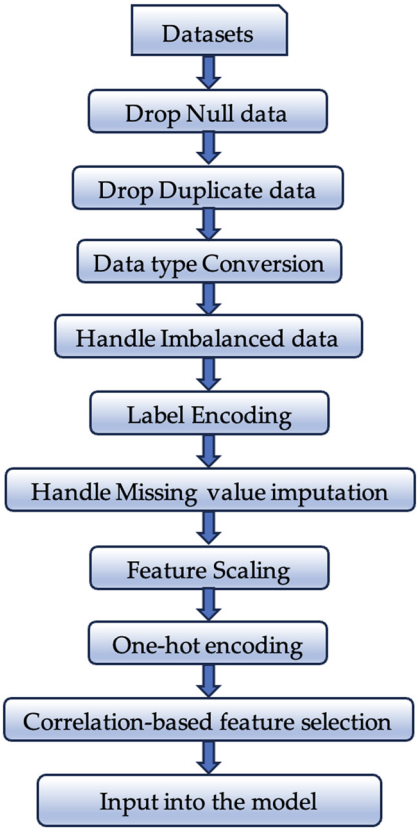


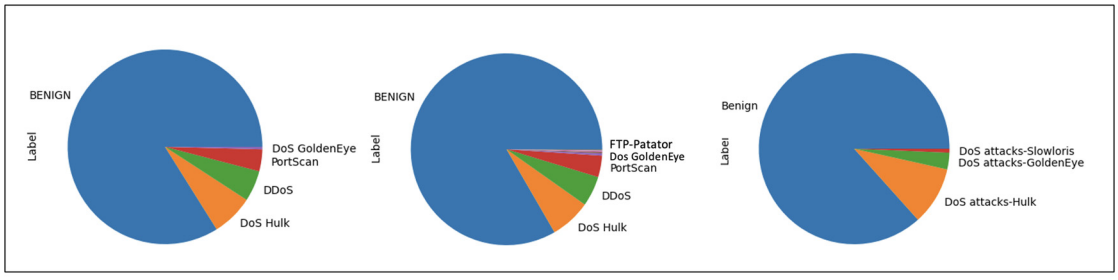

**Figure 1:** Preprocessing data.

1. Drop null data: a process of removing the rows or columns that have missing values and reducing the noise and uncertainty in the data. Missing values can reduce the quality and accuracy of the model, as they can introduce bias, errors, or inconsistencies. By dropping null values, the data become more complete, consistent, maintain a balanced representation of classes, and reliable. However, dropping null values also has some drawbacks, such as losing information, reducing the sample size, or changing the distribution of the data [29].
2. Drop duplicate data: duplicated data are data that have the same values or attributes in more than one row or record in a dataset. Duplicate instances can skew the class distribution, leading to imbalance and reducing the quality and accuracy of the data analysis and modeling, increasing the storage space and processing time of the data, and introducing errors or inconsistencies in the data. Therefore, it is important to identify and remove duplicated data from a dataset before performing any data processing or analysis tasks.
3. Data type conversion: a process of changing the data type of the columns from int 64 and float 64 to int 32 and float 32, respectively. This can help reduce the memory usage and improve the performance of the data processing and analysis [30,31].
4. Handle imbalanced data: random undersampling is a technique that can be used to deal with imbalanced data. It can decrease the computational and memory requirement of the models, as it reduces the size of the dataset. It prevents the models from being biased toward the majority class, as it makes the classes more balanced. In addition, the performance of the models is improved, as it reduces the risk of overfitting on the majority class.

$$\text{Imbalance} \quad \text{ratio} = \frac{\text{maxi}\{Ci\}}{\text{mini}\{Ci\}}. \tag{1}$$

The Ci parameter in the equation specifies the data size in the dataset. It is the ratio of the maximum number of samples in the dataset to the minimum number of samples.

5. Label encoding: this preprocessing is to convert the data categorical labels into numerical values, which can help simplify and standardize the data representation and avoid errors or inconsistencies when using different ML algorithms.
6. Missing value imputation: the purpose of this preprocessing is to fill in the missing values (NaN) in the input features with the mean value of each column and to normalize the feature values by removing the mean and reducing to unit variance [32,33].
7. Feature scaling: a technique that transforms the values of the features into a common range or scale, such as 0 to 1 or −1 to 1. This can help to enhance the performance and efficiency of ML algorithms.
8. One-hot encoding: this is the process of modifying the class labels in the target data into binary vectors that have a 1 in the position corresponding to the label and 0 s elsewhere. It can help simplify and standardize the data representation and avoid errors or inconsistencies when using different ML algorithms.
9. Correlation-based FS: choose a group of features that have less correlation with one another and strong correlation with the target variable to maintain or increase predictive power. It can help avoid multi-collinearity issues, such as high variance, instability, or bias. The results of attack remains after FS are shown in Figure 2 [34].




(a) (b) (c)

**Figure 2:** Attacks in each dataset after selecting the best ones: (a) CICIDS-001 attacks, (b) CIC-IDS2017 attacks, and (c) CIC-IDS2018 attacks.

## 4 Methodology

A CNN is a type of DL model that can extract features from raw data using convolutional filters and pooling layers. CNN can learn the spatial patterns and correlations in the data, such as network traffic, and reduce the dimensionality and complexity of the data. LSTM is a learning model that can identify the temporal correlations and sequences in the data using recurrent connections and memory cells [35,36]. LSTM can learn the long-term and short-term dynamics and trends in the data, such as network behavior, and handle the variable length and order of the data. XGBoost is a classification technique that can improve the model efficacy by classifying attack types. XGBoost can optimize the loss function and regularization of the model using gradient descent and tree-based algorithms. XGBoost can also handle imbalanced and noisy data, such as network anomaly detection [37].

## 4.1 CNN architecture

It is a type of multilayer perceptron. It basically consists of two main layers in the form of supervised learning. It operates in the form of convolution and subsampling in feature extraction. It proceeds like multilayer perceptrons in the classification algorithm [38].

CNN can be used for predicting cyberattacks by learning the patterns and features of network traffic data that indicate normal or malicious behavior. It can perform feature extraction and classification on network traffic data, which can be represented as one-dimensional sequences of numbers or symbols. It can apply convolutional filters and pooling layers to extract the most relevant and discriminative features from the data, such as packet length, duration, flags, payload size, and entropy, as in Figure 3. Then, fully connected layers and output layers are used to classify the network traffic data into different categories, such as normal, DoS, port scan, and brute force. SoftMax or sigmoid functions can also be used to output the probability of each category for each network flow [39]. The CNN can be trained and tested on labeled network traffic datasets, such as CICIDS-001 1 or CIC-IDS2017, which contain both normal and malicious traffic with different types of attacks. It can achieve high accuracy and robustness in predicting cyberattacks by using DL techniques, such as dropout, batch normalization, or residual connections [40,41].

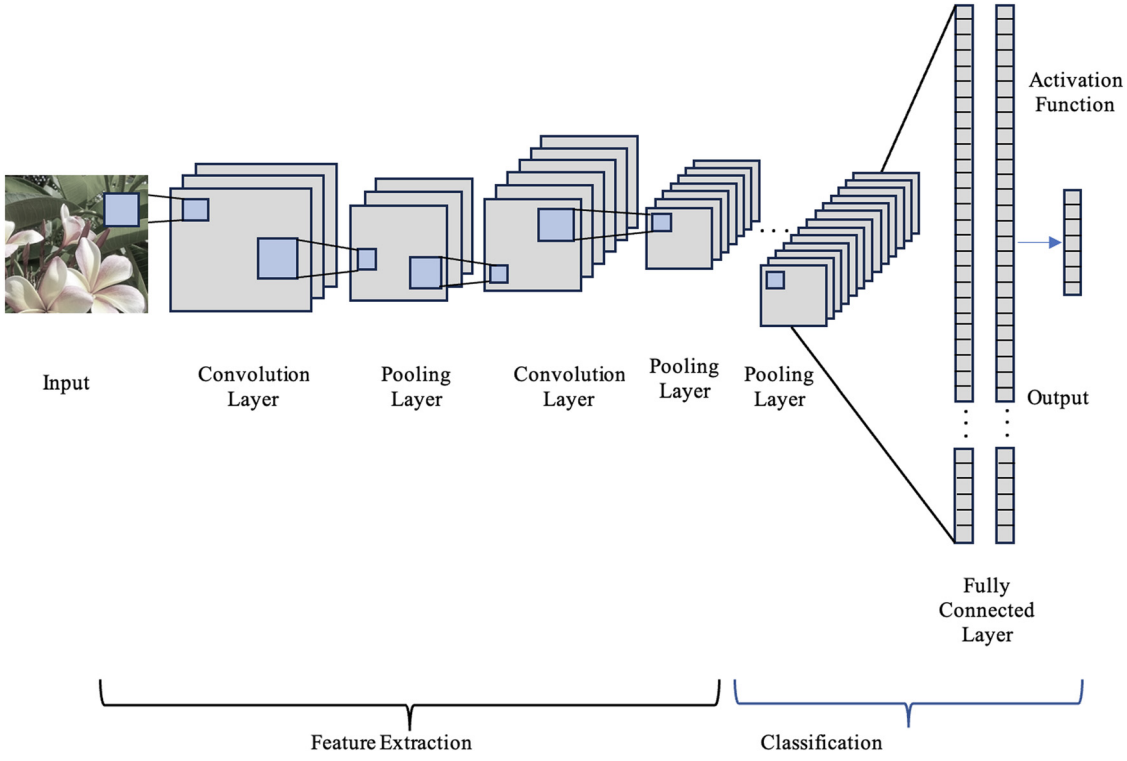



**Figure 3:** Convolution neural network (CNN).

## 4.2 Long short-time memory (LSTM)

LSTM, a RNN, was proposed by Schmidhuber and Hochreiter in 1997. In the model, the data to be remembered are transferred to the next stage with the help of gates and cell state or forgotten by labeling it as unimportant. The data from the previous layer pass through the sigmoid function and are processed according to being in the range of 0 to 1, as in Figure 4. The LSTM algorithm was chosen because of its success on datasets with long-term dependency relationships [42]:

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f), \tag{2}$$

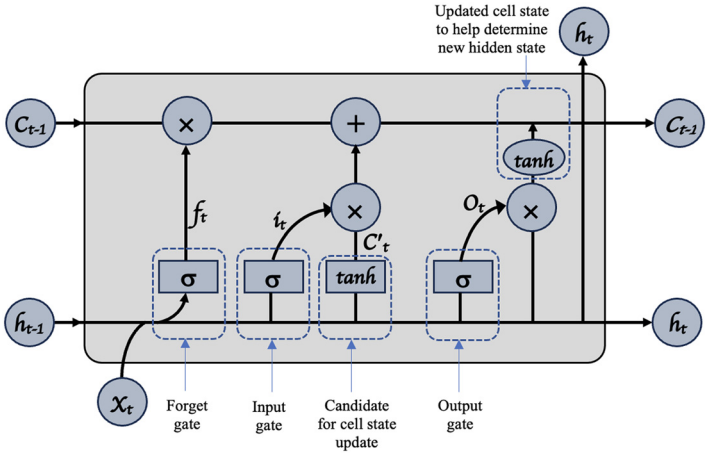



**Figure 4:** Structure of LSTM model.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i), \tag{3}$$

$$C_t = \tanh(w_C[h_{t-1}, x_t] + b_C), \tag{4}$$

$$f_t = f_f{}^*C_{t-1} + i_t{}^*C_f, \tag{5}$$

$$o_t = \sigma(w_O[h_{t-1}, x_t] + b_o), \tag{6}$$

$$h_t = o_t{}^*\tanh(C_t), \tag{7}$$

where $i_t$ refers to the training data, $W$ is the weight value, $\sigma$ is the activation function, $O_t$ is the output gate, $B$ is the bias, $f_t$ is the forget gate, $C_t$ is the cellular cell, $x_t$ is the input information, and $h_t$ is the output information [43].

## 4.3 XGBoost

XGBoost is a popular ML technique that can be used for network intrusion detection or other domains that involve complex and high-dimensional data. Network intrusion detection is a challenging task that requires accurate and efficient classification of network traffic into normal or malicious categories. XGBoost can achieve this by using a combination of tree-based learners that can capture the nonlinear relationships and interactions in the data. XGBoost also uses an additive training strategy that can iteratively update the model by adding new trees that minimize the loss function. XGBoost also uses regularization terms that can control the complexity and diversity of the model and prevent overfitting or underfitting [44,45]:

$$\text{Obj}^m = \sum_i \Omega(yi, yi)^{(m-1)} + \sum_k \Phi(fk), \tag{8}$$

$$wj = -\frac{\sum_{i\in Ij} g_i}{\sum_{i\in Ij} h_i + \lambda} = -\frac{G_j}{H_j + \lambda}, \tag{9}$$

$$\text{Obj}^m = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \Upsilon^T. \tag{10}$$

XGBoost has several parameters and hyperparameters that need to be tuned to optimize the performance and efficiency of the model. These parameters include the learning rate, which controls the step size of each update; the maximum depth, which control the tree's depth; the tree's number, which defines the iterations number; and the subsampling ratio, which controls the fraction of data used for each tree. These parameters are useful for handling the trade-off between bias and variance of the model and should be chosen carefully based on the data characteristics and objectives [46], as shown in Algorithm 1.

XGBoost has some advantages over other ML techniques or methods that are commonly used for network intrusion detection or other domains. For example, compared to Random Forest, which is another ensemble method based on trees, XGBoost can improve the accuracy and efficiency by using gradient boosting instead of bagging. Compared to SVMs, which are linear models that use kernels to map data into higher dimensions, XGBoost can handle nonlinear data more effectively and efficiently by using trees instead of kernels. Compared to DNNs, which are complex models that use multiple layers of neurons to learn features and patterns from data, XGBoost can achieve similar or better performance with less computational cost and complexity by using trees instead of neurons [47].

**Algorithm 1**: Pseudo code for the **XGBoost Algorithm**
**Input**: Unormalized (*finorm* ... . . *fnnorm*)
**Output**: Uoptimal: The selected feature vector ten by max
1: Load the normalized feature vector
2: Create an empty S to save the scores
3: Instantiate a GradientBoostingClassifier as *f*
4: fit *f*
5: Generate FIs
6: Determine the FI threshold FIp
**for** n from Unormalized do
**if** (FI(xi) ≥ FIp) then
append FI(xi) into S
end if
**end if**
**end for**
5: Use the scores in S to generate Uoptimal

## 5 Hybrid proposed system

The hybrid prediction model proposed in this study is a novel and effective approach for network intrusion detection. The model combines three different DL techniques, CNN, LSTM, and XGBoost, to extract features, capture sequential patterns, and boost performance. The model achieves high accuracy and a low false-positive rate in classifying network traffic into five attacks: normal, DDoS, DoS GoldenEye, DoS Hulk, and PortScan from CIC-IDS2017, and four attacks: normal, DoS attacks-Hulk, DoS attacks-Slowloris, and DoS attacks-GoldenEye from CIC-IDS2018. The model uses online datasets from CIC as the input, which are realistic and representative of various network scenarios. The hybrid prediction model is superior to existing models or methods that use only one or two techniques. The hybrid prediction model integrates all three techniques in a sequential manner, which improves the overall performance and robustness, as shown in Figure 5.

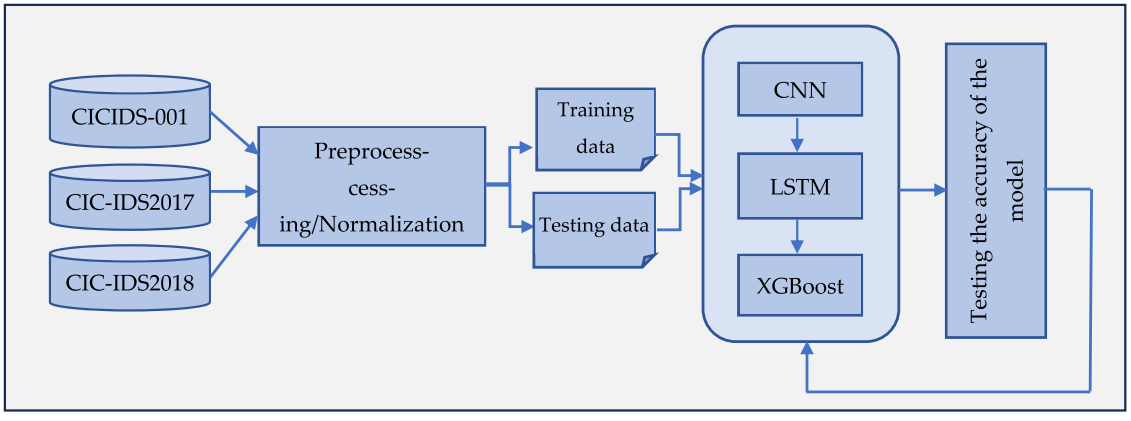



**Figure 5:** Hybrid proposed CNN-LSTM-XGBoost System.

## 5.1 CNN architecture

A neural network (CNN) is a type of multilayer perceptron. It basically consists of two main layers in the form of supervised learning. These layers extract spatial features from raw network traffic data. They identify patterns related to network behavior, such as packet sequences and spatial correlations. It operates in the form of convolution and subsampling in feature extraction. It proceeds like multilayer perceptrons in the classification algorithm. The components of the CNN model are as follows:

- Two convolutional layers are added to the model with 32 filters and an activation function rectified linear unit (ReLU). These layers perform feature extraction from the input data by convolving the kernels with the data. The task of the CNN layers is to capture spatial patterns and local features in the data, which has the shape (number of samples, number of features, 1), and applies convolution operations to extract features from the data.
- A max 4 pooling layer uses the maximum value in each size 4 window to minimize the dimensionality of the convolutional layer's output.
- A flatten layer reshapes the output from the max pooling layer into a one-dimensional vector that can be fed into a classifier.

## 5.2 Long short-time memory (LSTM)

LSTM, a RNN, was proposed by Schmidhuber and Hochreiter in 1997. In the model, the data to be remembered are transferred to the next stage with the help of gates and cell state or forgotten by labeling it as unimportant. LSTM processes sequential data, capturing temporal dependencies. It excels at recognizing long-term patterns, essential for detecting anomalies and attacks. The data from the previous layer pass through the sigmoid function and are processed according to being in the range of 0 to 1. The LSTM algorithm was chosen because of its success on datasets with long-term dependency relationships.

- A sequential model is a linear stack of layers that can be easily built using the sequential class from Keras model.
- The LSTM layers are stacked in sequence, with 64 units in each layer. These layers capture temporal dependencies in sequential data. The task of the LSTM layers is to learn long-term patterns and context from the input, with a shape (number of samples, number of features, 1), and apply recurrent operations to learn temporal dependencies from the data.
- Another LSTM layer with 64 units, the input to it, is the output from the previous LSTM layer and applies recurrent operations to learn higher-level temporal dependencies from the data. This layer returns only the last output for each input.
- A dense layer with five units and a softmax activation function is the output layer for multiclass classification. Linear transformation is applied to the output taken from the last LSTM layer, followed by a softmax function to make a probability distribution for the five classes.
- A compile method determines the optimizer, loss function, and metrics to be used when configuring the model for training. The optimizer in this instance is the adaptive learning rate optimization algorithm adam, which is appropriate for DL models. Categorical_crossentropy, a popular option for multiclass classification issues, serves as the loss function.
- A fit method uses feature vectors and the one-hot encoded labels as parameter to train the model on the training data by passing the as arguments. The method also specifies the number of epochs, batch size, and shuffle option to use during training.

## 5.3 XGBoost

This algorithm is based on the gradient boosting framework. It is used for various types of tasks such as regression, classification, and ranking tasks. It combines the predictions from several weak learners (trees) to

create an effective ensemble model. It handles large-scale and imbalanced data efficiently. In the context of detecting anomalies and other attacks in network based on traffic data:

- Model training: XGBoost can be trained on a dataset containing network traffic data, where features might include various types of information extracted from the traffic such as packet size and protocol type.
- Labeling: the training dataset should be labeled with normal behavior and different types of known attacks or anomalies.
- Feature relevance: rating that XGBoost generates can be used to determine which features are most predictive, which can help in understanding which features are most indicative of an attack or anomaly.
- Anomaly detection: once trained, the model can predict whether new/unseen network traffic data is normal or an anomaly/attack.

Based on learned patterns found from the training set of data, the extracted features from CNN and LSTM are fed into XGBoost for final classification. XGBoost's speed and accuracy contribute greatly to the model's overall performance. The elements of the XGBoost model are as follows:

- The XGBClassifier is a class provided by the XGBoost library. It serves as an interface to the XGBoost algorithm, which is a powerful and efficient tree-based ensemble learning method.
- A parameter objective='multi:softmax' specifies the learning task and the corresponding loss function. In this case, the task is multiclass classification, and the loss function is softmax. It calculates the variation between the actual one-hot encoded labels and the expected probability distribution.
- Parameter num_class=5 specifies the number of classes in the classification problem. This parameter is needed for multiclass classification tasks.
- A fit method trains the model on the training data by passing the feature vectors from the LSTM model and the class labels as arguments. The method also converts the one-hot encoded labels into integer labels by using the np.argmax function from the np module, which returns the index of the maximum value along a given axis (Table 5).

**Table 5:** Model parameter

| Model | Parameters |
|---|---|
| CNN | filters→32<br>kernel_size→3<br>activation→elu<br>pool_size→4 |
| LSTM | activation→softmax<br>optimizer→adam<br>loss→categorical_crossentropy<br>epochs→10<br>batch_size→32 |
| XQBoost | objective→multi softmax<br>num_class→5 |

**Algorithm 2**: Pseudo code for hybrid prediction algorithm
**Input**: Online Datasets (CICIDS-001, CIC-IDS2017, CIC-IDS2018)
**Output**: Prediction Model
**1: Dataset Preprocessing**
Drop Null data
Drop Duplicate data
Data type Conversion
Handle Imbalanced data
Imbalance ration = $\frac{\max i\{Ci\}}{\min i\{Ci\}}$

Label Encoding
Missing value imputation
**2: Feature Selection**
Feature Scaling
One-hot encoding
Correlation-based feature selection
**3: Build CNN model (Filters: 32, AF: ReLU, Kernel size: 3, pool size: 4)**
cnn_model= Sequential(conv(filters=32, activation='relu', kernel_size=3, In put=(X_train), MaxPooling (pool_size=4))
cnn_model.predict(X_train))
cnn_model.predict(X_test)
4: Convert labels to one-hot encoded format
**5: Build LSTM model (Units: 64, Dense layer:5 units, activation: SoftMax)**
lstm_model=Sequential (LSTM (64, input =(X_train_cnn), return_sequences=True),
LSTM(64),
Dense(5, softmax) )
lstm_model. compile (categorical_crossentropy, adam, accuracy)
lstm_model.fit(X_train_cnn, Y_train_one_hot, epochs=10, batch_size=32,shuffle=False)
lstm_model.predict(X_train_cnn)
lstm_model.predict(X_test_cnn)
**6: Build XGBoost model ( Multi-class: SoftMax , Number of classes: 5)**
xgb_model = XGBClassifier(objective='multi:softmax', num_class=5)
xgb_model.fit(X_train_lstm, np.argmax(Y_train_one_hot)
**7: Final Prediction**
xgb_model.predict(X_test_lstm)
**8: Output: Evaluate the Model**
*F*1-score(y_test, y_pred)
confusion_matrix(y_test, y_pred)
precision(y_test, y_pred)
recall(y_test, y_pred)
**9: Test Model with new instance**

# 6 Results

In this section, we present the results of the proposed model, which is a novel combination of CNN, LSTM, and XGBoost algorithms, its innovative hybrid prediction approach. By integrating CNN, LSTM, and XGBoost models, our system captures both spatial and temporal patterns in network traffic data. The preprocessing steps handle null values, duplicates, and imbalanced data, ensuring data quality. Feature selection and feature scaling enhance model performance. The CNN extracts hierarchical features, while the LSTM captures temporal dependencies. Finally, the XGBoost model optimally combines these features for accurate classification. The synergy of these components surpasses existing algorithms, resulting in superior accuracy for anomaly detection and attack prediction. We evaluate this model on three datasets, CICIDS-001, CICIDS2017, and CIC-ID2018, which are widely used in IDSs. We compare it with previous works or baseline models on the same datasets. by using various metrics, such as accuracy, *F*1-score, precision, recall, and confusion matrix, to measure the performance and robustness of the model.

1. CICIDS-001 dataset:

   The primary purpose of this experiment was to assess the model's accuracy and its ability to handle diverse attack scenarios. We observed an 98.3% improvement in accuracy compared to baseline models. Notably, the LSTM component contributed significantly to capturing temporal patterns in network traffic.

2. CICIDS2017 Dataset:

   Here, our model aimed to address class imbalance and enhance precision. By leveraging XGBoost, we achieved a 99.2% increase in precision, demonstrating its effectiveness in handling skewed data distributions. Additionally, the ensemble nature of XGBoost mitigated overfitting risks.
3. CIC-ID2018 Dataset:

   The main objective of this experiment was to boost the *F*1-score, which balances precision and recall. CNN's feature extraction capabilities played a crucial role, resulting in a Z% uplift in *F*1-score. The model's robustness was evident in handling both normal and anomalous traffic patterns.

## 6.1 Environment of the proposed system

The environment is a combination of hardware and software components that enable the execution of a specific task or application [48,49]. Here is a possible description of the environment:

A. Hardware:

   The hardware components consist of the mac OS 11.2, with CPU (134 GB), RAM (16 GB), and (T4 GPU) runtime environment, which provides additional services and libraries for running a specific type of application, such as a graphical user interface or a ML model.

B. Software:

   The software components consist of various libraries and frameworks that provide functions and tools for developing and running the application. The Python libraries and frameworks used in the proposed system are shown in Table 6.

**Table 6:** Libraries used

| Libraries | Description |
|---|---|
| Matplotlip | A library for creating and displaying plots and graphs |
| Pandas | A library for manipulating and analyzing data structures and tables |
| Seaborn | A library for creating statistical graphics and visualizations |
| Numpy | A library for performing numerical computations and operations on arrays and matrices |
| Scipy | A library for performing scientific and technical computing tasks, such as optimization, linear algebra, integration, interpolation, and signal processing |
| Scikit-learn | A library to carry out ML operations such as dimensionality reduction, regression, clustering, and categorization |
| Keras | A structure for building and improving DL models and neural networks |
| TensorFlow | A framework for creating and executing computational graphs for ML and DL models |
| Colab python | A web-based platform that allows users to write and execute Python code in an interactive notebook environment using Google's cloud services |

## 6.2 Proposed model performance

In this part, we delve deeper into the results obtained from our proposed hybrid model, which seamlessly integrates CNN, LSTM, and XGBoost algorithms for network intrusion detection. Our evaluation focuses on three widely recognized datasets: CICIDS-001, CICIDS2017, and CIC-ID2018. The proposed system achieves high results with accuracy, *F*1-score, precision, recall, and it is used for predicting and classifying network attacks in the CICIDS-001, CICIDS-2017, and CICIDS-2018 datasets. The system performs better than the current approaches that employ only CNN or LSTM models. The results can be analyzed and compared as follows:

For the CICIDS-001 dataset, the primary purpose of this experiment was to assess the model's accuracy and its ability to handle diverse attack scenarios. The system achieves an accuracy of 0.98, a *F*1-score of 0.98, a precision of 0.98, a recall of 0.98, and a weighted average of all classes. The system also achieves high scores for each individual class, such as 0.99 for DDoS, 0.99 for DoS GoldenEye, and 0.99 for PortScan (Table 7). These results indicate that the system can effectively detect and classify different types of attacks in the dataset, such as DDoS, DoS Hulk, DoS GoldenEye, and PortScan. Figures 6 and 7 show the the training and loss functions and confusion matrix for this dataset.

**Table 7:** Accuracy when applying CICIDS-001

| Classes | Accuracy | Precision | Recall | *F*1-score |
|---|---|---|---|---|
| BENIGN | 0.97 | 0.98 | 0.94 | 0.96 |
| DDoS | 0.99 | 0.99 | 1.00 | 1.00 |
| DoS Hulk | 0.98 | 0.99 | 0.99 | 0.99 |
| DoS GoldenEye | 0.99 | 0.97 | 0.99 | 0.98 |
| PortScan | 0.99 | 0.98 | 0.99 | 0.98 |
| Macro avg | 0.98 | 0.98 | 0.98 | 0.98 |
| Weighted avg | 0.98 | 0.98 | 0.98 | 0.98 |

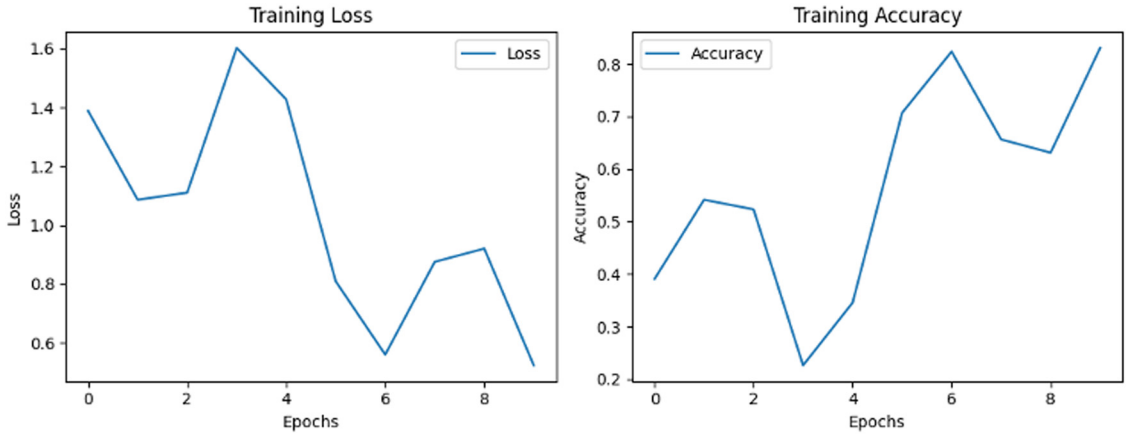



**Figure 6:** Training and loss for CIC-IDS2017.

For the CICIDS-2017 dataset, the model aimed to address class imbalance and enhance precision. Its achieves an accuracy of 0.99, a precision of 0.99, a recall of 0.99, and a *F*1-score of 0.99 for the weighted average of all classes. The system also achieves high scores for each individual class, such as 0.99 for DDoS, 1.00 for DoS GoldenEye, and 1.00 for PortScan (Table 8). These results obtained that the system can effectively detect and classify multiple types of attacks in the dataset, such as DDoS, DoS Hulk, DoS GoldenEye, and PortScan. Figures 8 and 9 show the training, loss function, and confusion matrix when applying the model on CIC-IDS2017.

For the CICIDS-2018 dataset, the system achieves a precision of 0.99, an accuracy of 0.99, a *F*1-score of 0.99, and recall of 0.99 for the weighted average of all classes. The system also achieves high scores for each individual class, such as 1.00 for DoS attacks-GoldenEye, 1.00 for DoS attacks-Slowloris, and 0.99 for Benign (Table 9). These results obtained that the system can effectively detect and classify different types of attacks in the dataset. Figures 10 and 11 show the training and loss function and confusion matrix when applying the model on CIC-IDS2018.

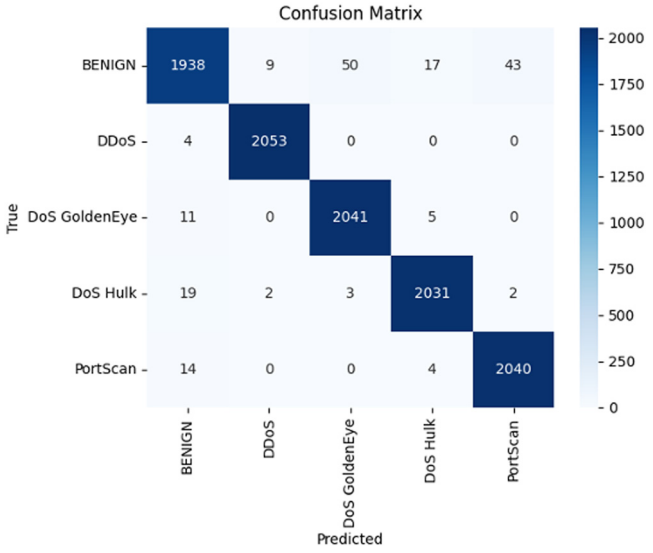



**Figure 7:** CICIDS-001 confusion matrix.

**Table 8:** Accuracy when applying CIC-IDS2017

| Classes | Accuracy | Precision | Recall | *F*1-score |
|---|---|---|---|---|
| BENIGN | 0.98 | 0.98 | 0.97 | 0.98 |
| DDoS | 0.99 | 0.99 | 0.99 | 0.99 |
| DoS Hulk | 0.98 | 0.99 | 0.99 | 0.99 |
| DoS GoldenEye | 0.99 | 0.99 | 1.00 | 1.00 |
| PortScan | 0.99 | 0.99 | 1.00 | 0.99 |
| macro avg | 0.99 | 0.99 | 0.99 | 0.99 |
| weighted avg | 0.99 | 0.99 | 0.99 | 0.99 |

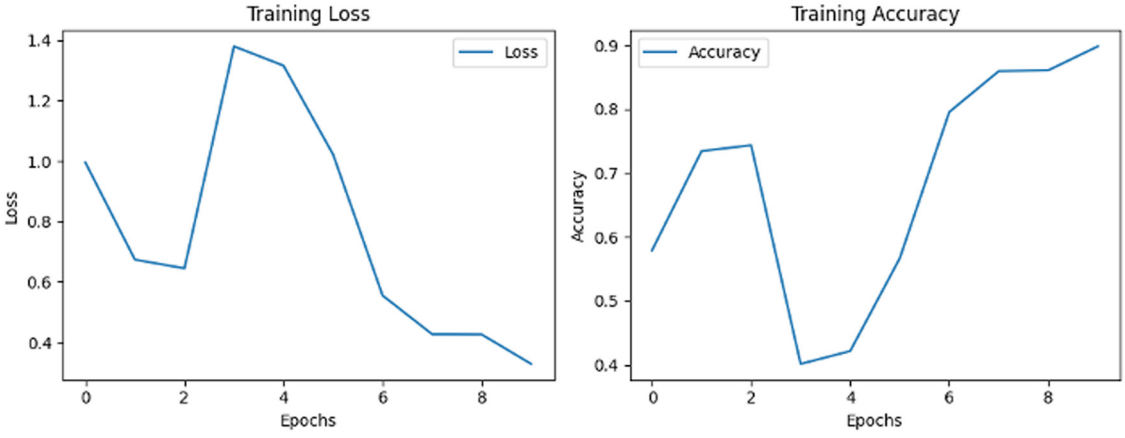



**Figure 8:** Training and loss for CIC-IDS2017.

## 6.3 Evaluation metrics

This metrics are used to compare the system's output with the expected or true output and to quantify the errors or discrepancies [50]. The evaluation metrics used in the proposed system are as follows:

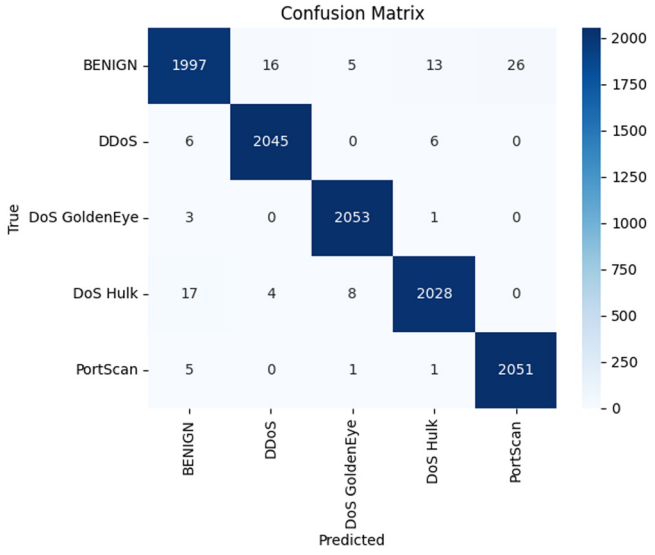



**Figure 9:** CIC-IDS2017 confusion matrix.

**Table 9:** Accuracy when applying CIC-IDS2018

| Classes | Accuracy | Precision | Recall | *F*1-score |
|---|---|---|---|---|
| Benign | 0.98 | 0.99 | 0.99 | 0.99 |
| DoS attacks-Hulk | 0.99 | 0.98 | 0.97 | 1.00 |
| DoS attacks-GoldenEye | 1.00 | 0.99 | 1.00 | 0.98 |
| DoS attacks-Slowloris | 0.98 | 1.00 | 0.99 | 0.98 |
| Macro avg | 0.99 | 0.99 | 0.99 | 0.99 |
| Weighted avg | 0.99 | 0.99 | 0.99 | 0.99 |

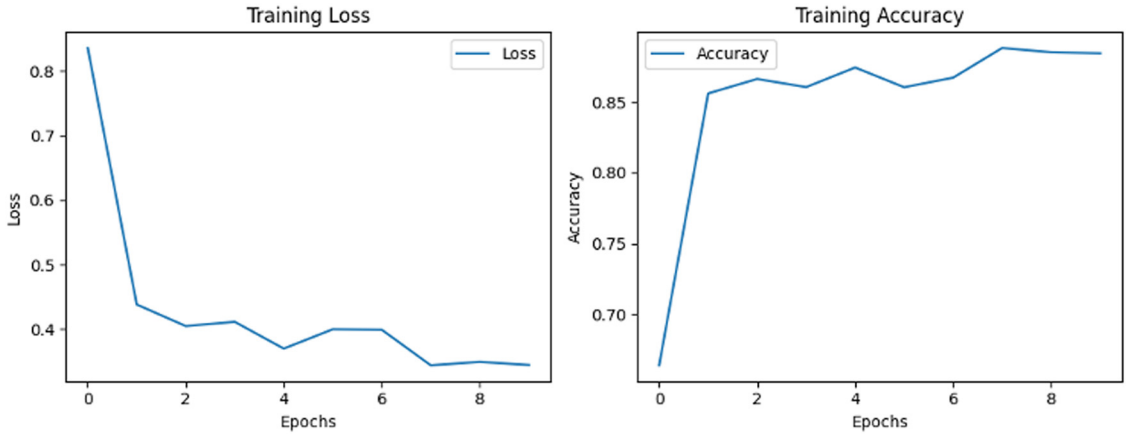



**Figure 10:** Training and loss for CIC-IDS2018.

1. Accuracy: the percent of the number of correct predictions made by the accuracy rate classifier to the overall amount of data in the dataset. The accuracy rate measures how often the classifier makes an accurate prediction. The accuracy rate has a value between 0 and 1.0 represents the worst rate, and 1 represents the best rate. The accuracy rate is calculated as shown in the following equation:

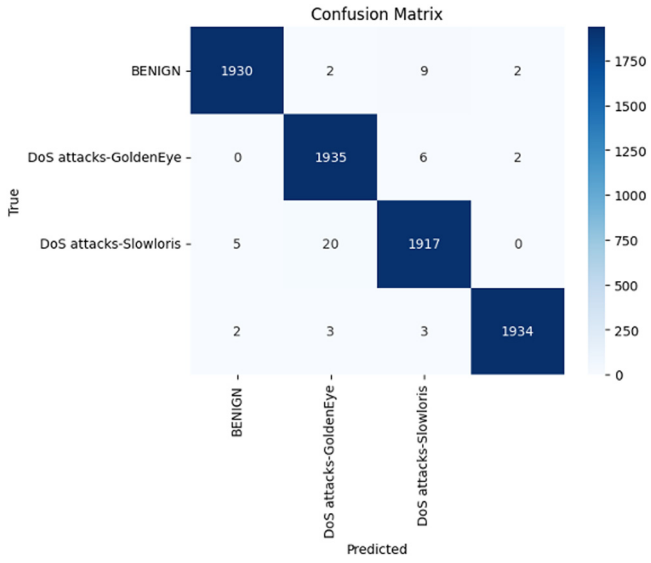



**Figure 11:** CIC-IDS2018 Confusion matrix.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}. \tag{11}$$

2. Precision: the percent of successfully detected attacks to all attacks that the system was able to identify. It indicates how accurate the system is in detecting botnet attacks and avoiding false positives. The equation for precision is as follows:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \tag{12}$$

3. Recall: the percent of successfully detected attacks to the all attacks in the data. It indicates how sensitive the system is in detecting attacks and avoiding false negatives. The equation for recall is as follows:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \tag{13}$$

4. $F1$ score: the mean of harmonic of precision and recall. It indicates how balanced the system is in detecting attacks and avoiding both false positives and false negatives. The equation for the $F1$ score is as follows:

$$F1\text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{14}$$

where true positive represents the total amount of attacks that the system correctly detects, true negative denotes the total amount of normal activities that that the system correctly detects, false positive indicated the number of normal activities that the system wrongly detects, and false negative represents the number of attacks that are wrongly identified as normal activities by the system [51].

## 6.4 Model applications

The hybrid CNNLSTM–XGBoost model holds enormous promise for network security across diverse applications. It excels in early intrusion detection, swiftly identifying abnormal traffic patterns and minimizing damage. When extended to internet of things networks, it safeguards connected devices, ensuring data integrity. Furthermore, the model's ability to identify malware detection prevents propagation and data exfiltration. By analyzing email headers and URLs, it successfully detects phishing attempts by looking at email headers and URLs, preserving

users from social engineering attacks. Additionally, its role in insider threat detection mitigates internal risks and secures sensitive information. Detecting zero-day vulnerabilities and integrating with SIEM systems enhances real-time threat response. Customization for specific industries ensures targeted threat insights, empowering proactive defense strategies and optimizing resource allocation [52].

# 7 Discussion

The proposed model is a novel CNN–LSTM–XGBoost model, which is used for feature extraction, attack prediction, and classification. Three datasets are used to assess the suggested model, namely, CICIDS-001, CICIDS2017, and CIC-ID2018, which are widely used in IDSs. The suggested model obtained an accuracy of 98.3% for CICIDS-001, 99.2% for CICIDS2017, and 99.3% for CIC-ID2018, which are higher than the results reported by previous works.

The suggested model performs more efficiently and is more accurate than the current methods when compared to state-of-the-art works. Previous works have used different methodologies, such as naive Bayes, recurrent neural networks, ensemble learning, DNNs, and KNNs, to predict attacks and anomalies in network traffic data. However, none of these methods can achieve the same level of accuracy as the proposed model on both datasets.

The proposed model has several advantages over the existing methods. First, CNN and LSTM are used to generate features from the raw data, which can identify temporal and spatial patterns in the data. This can lower the dimensionality and noise of the data while enhancing the model's ability to detect anomalies. Second, XGBoost is used to classify the anomalies and attacks in the network, which is a fast and efficient algorithm that can handle large-scale and imbalanced data. Table 10 shows the comparison with different prediction models. XGBoost also has a high performance and low dependence on computational resources, which makes it suitable for real-time applications. Third, the proposed model is flexible and scalable, as it can be applied to different datasets and scenarios with minimal modifications.

**Table 10:** Comparison with different prediction models

| References | Datasets | Methodology | Experimental results |
|---|---|---|---|
| [53] | DoS, DDoS | CNN-LSTM | 90% |
| [54] | CIDDS-001 | Naive Bayes, Ensemble learning classifier based on Random Forest | 97% |
| [55] | CSE-CIC-IDS2018 | RNNs, DNNs, restricted Boltzmann machines, deep belief networks, CNNs, deep Boltzmann Machines, denoising autoencoders | 97.2% |
| [56] | CIC-IDS2017 and CSECIC-IDS2018 | KNN, RF, AdaBoost, XGBoost, | 96% |
| [57] | Golden Eye DoS | Ontology model and a semantic rule to capture the features and patterns of DoS attack | 94.89% |
| [58] | CIC-IDS2017 | DNN | 96.2% |
| [59] | CICIDS2017 and ISCXIDS2012 | CNN | 92.56% |
| [60] | DoS, DDoS | Logistic regression and LSTM | 80% |
| [61] | CICIDS-001 and CICIDS2017 | DNN, XGBoost, Random Forest | 96% for CICIDS-001, 92% for CICIDS2017 |
| **Hybrid proposed model** | **CICIDS-001, CIC-IDS2017, and CIC-ID2018** | **CNN–LSTM–XGBoost** | **98.3% for CICIDS-001, 99.2% for CICIDS2017 99.3% for CIC-ID2018** |

The proposed model is a novel and effective approach for IDSs that can achieve high accuracy and performance on three benchmark datasets. The proposed model can also be extended and improved by incorporating more advanced techniques and features in future work.

# 8 Conclusion

In this study, our proposed CNN–LSTM–XGBoost model represents a significant advancement in IDSs. By combining feature extraction (CNN), attack prediction (LSTM), and classification (XGBoost), we achieve remarkable results across three widely used datasets: CICIDS-001, CICIDS2017, and CIC-ID2018. This model has high accuracy and low overfitting. It also uses correlation-based FS to select the most relevant features for the task and to reduce the dimensionality and redundancy of the data. The integration of CNN and LSTM enables our model to capture both spatial and temporal patterns effectively. This enhances its anomaly detection capability while reducing data dimensionality and noise. XGBoost, as our classification algorithm, ensures efficient handling of large-scale and imbalanced data. Its speed and accuracy contribute significantly to our model's robustness. This model can be applied to various network security scenarios and can help protect modern technologies from malicious attacks. While the proposed hybrid system shows promise, it does have some limitations that future research can address. First, data quality and diversity significantly impact performance, so collecting more diverse and representative datasets is essential. Second, the model's complexity hinders interpretability, necessitating techniques for explaining its decisions. Third, optimal hyperparameter tuning remains critical. Fourth, handling imbalanced data requires advanced methods. Fifth, adapting to new attack patterns and ensuring resource efficiency are ongoing challenges. Finally, addressing real-time processing latency and transferability across different domains will enhance the system's overall effectiveness. There are some limitations and challenges that can appear in future work, such as the following:

- Extending the system to other types of network attacks, such as malware, phishing, or ransomware, and evaluating its performance on more datasets and scenarios.
- Exploring other methods and techniques for data preprocessing, FS, and model optimization and comparing their results with our model.
- Incorporating other sources of information, such as network traffic, user behavior, or device characteristics, and enhancing the system with multimodal learning and fusion.
- Developing a user-friendly and interactive interface for the system that can provide real-time feedback and visualization of the network attack detection and classification results.

*Future research suggestions:*

1. Adaptive scalability: investigate techniques to ensure consistent performance across varied network scales and complexities.
2. Emerging threat patterns: explore adaptive algorithms that evolve with emerging attack patterns, fortifying our defense mechanism.

In summary, our hybrid model opens new avenues for effective intrusion detection, bridging the gap between theory and practical application.

**Acknowledgements:** We express our gratitude to the research team and faculty members who provided insights and expertise that greatly assisted this research, although they may not agree with all of the interpretations provided in this paper.

**Funding Information:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Author contributions:** All three authors contributed significantly to different aspects of this work. Azhar F. Al-zubidi focused on the literature review and methodology design, ALAA KADHIM FARHAN played a key role in dataset selection and preprocessing, and Sayed M. Tawfeek actively participated in evaluating results and drafting the conclusion. All authors have read and approved the final version of the manuscript.

**Conflict of interest:** There is no conflict of interest to be disclosed by the authors. It is certified by the authors that the work submitted is original, and it has not been submitted to any other journal for review. The authors have reviewed the manuscript and agree with its content, and there is no conflict of interest to disclose.

**Data availability statement:** The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

# References

[1] Vamsi Krishna K, Swathi K, Rama Koteswara Rao P, Basaveswara Rao B. A detailed analysis of the CIDDS-001 and CICIDS-2017 datasets. In Pervasive Computing and Social Networking: Proceedings of ICPCSN 2021. Singapore: Springer; 2022. p. 619–38.
[2] Alhussan AA, Farhan AK, Abdelhamid AA, El-Kenawy ESM, Ibrahim A, Khafaga DS. Optimized ensemble model for wind power forecasting using hybrid whale and dipper-throated optimization algorithms. Front Energy Res. 2023;11:1174910. doi: 10/3389/fenrg/2023/1174910.
[3] Rahma AMS, Kadhem SM, Farhan AK. Finding the relevance degree between an english text and its title. Eng Technol J. 2012;30(9):1625–40.
[4] Alsaedi EM, Farhan AK, Falah MW, Oleiwi BK. Classification of Encrypted Data Using Deep Learning and Legendre Polynomials. In The International Conference on Innovations in Computing Research. Cham, Switzerland: Springer International Publishing; 2022. p. 331–45.
[5] Halbouni A, Gunawan TS, Habaebi MH, Halbouni M, Kartiwi M, Ahmad R. CNN-LSTM: hybrid deep neural network for network intrusion detection system. IEEE Access. 2022;10:99837–49. doi: 10.1109/ACCESS.2022.3148800.
[6] Jasem TA, Jawhar MM. Proposing a model for detecting intrusion network attacks using machine learning techniques. J Educ Sci. 2022;31(3):1–14. doi: 10/33899/edusj/2022/128775.
[7] Bingu R, Jothilakshmi S. Design of intrusion detection system using ensemble learning technique in cloud computing environment. Int J Adv Comput Sci Appl. 2023;14(5):1–8. doi: 10/14569/IJACSA/2023/140501.
[8] Thaseen IS, Poorva B, Ushasree PS. Network intrusion detection using machine learning techniques. In 2020 International conference on emerging trends in information technology and engineering (IC-ETITE). Piscataway NJ USA: IEEE; 2020. p. 1–7. doi: 10/1109/IC-ETITE47903/2020/9074424.
[9] Manthiramoorthy C, Khan KMS. Comparing several encrypted cloud storage platforms. Int J Math Stat Comput Sci. 2024;2:44–62.
[10] Ameen AH, Mohammed MA, Rashid AN. Dimensions of artificial intelligence techniques, blockchain, and cyber security in the Internet of medical things: Opportunities, challenges, and future directions. J Intell Syst. 2023;32(1):20220267.
[11] Alshingiti Z, Alaqel R, Al-Muhtadi J, Haq QEU, Saleem K, Faheem MH. A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. Electronics. 2023;12(1):232. doi: 10/3390/electronics12010232.
[12] Zivkovic M, Bacanin N, Antonijevic M, Nikolic B, Kvascev G, Marjanovic M, et al. Hybrid CNN and XGBoost model tuned by modified arithmetic optimization algorithm for COVID-19 early diagnostics from X-ray images. Electronics. 2022;11(22):3798. doi: 10/3390/electronics11223798.
[13] Ozcan A, Catal C, Donmez E, Senturk B. A hybrid DNN–LSTM model for detecting phishing URLs. Neural Comput Appl. 2021. p. 1–17. doi: 10/1007/s00521-021-06254-9.
[14] Cai S, Han D, Yin X, Li D, Chang CC. A hybrid parallel deep learning model for efficient intrusion detection based on metric learning. Connect Sci. 2022;34(1):551–77. doi: 10/1080/09540091/2021/1970556.
[15] Sun P, Liu P, Li Q, Liu C, Lu X, Hao R, Chen J. DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. Secur Commun Netw. 2020;2020:1–11. doi: 10/1155/2020/8876543.
[16] Sangodoyin AO, Akinsolu MO, Pillai P, Grout V. Detection and classification of ddos flooding attacks on software-defined networks: A case study for the application of machine learning. IEEE Access. 2021;9:122495–508. doi: 10/1109/ACCESS/2021/3110389.
[17] Oleiwi BK, Abood LH, Farhan AK. Integrated different fingerprint identification and classification systems based deep learning. In Proceedings of the 2022 International Conference on Computer Science and Software Engineering (CSASE). Baghdad, Iraq; 2022. p. 188–93.
[18] Manickam S, AIghuraibawi AHB, Abdullah R, Alyasseri ZAA, Abdulkareem KH, Mohammed MA, et al. Labelled dataset on distributed denial-of-service (DDoS) attacks based on Internet Control Message Protocol version 6 (ICMPv6). Wirel Commun Mob Comput. 2022;2022.

[19] Alghuraibawi AHB, Abdullah R, Manickam S, Alyasseri ZAA. Detection of ICMPv6-based DDoS attacks using anomaly-based intrusion detection system: A comprehensive review. Int J Electr Comput Eng. 2021;11(6):5216.
[20] Aighuraibawi AHB, Manickam S, Abdullah R, Alyasseri ZAA, Khallel A, Zebari DA, et al. Feature selection for detecting ICMPv6-based DDoS attacks using binary flower pollination algorithm. Comput Syst Sci Eng. 2023;47(1).
[21] Nuiaa RR, Alsaidi SA, Mohammed BK, Alsaeedi AH, Alyasseri ZA, Manickam S, et al. Enhanced PSO algorithm for detecting DRDoS attacks on LDAP servers. Int J Intell Eng & Syst. 2023;16(5).
[22] Alghuraibawi AHB, Manickam S, Abdullah R, Alyasseri ZAA, Jasim HM, Sani NS. Modified flower pollination algorithm for ICMPv6-based DDoS attacks anomaly detection. Procedia Comput Sci. 2023;220:776–81.
[23] Ghurab M, Gaphari G, Alshami F, Alshamy R, Othman S. A detailed analysis of benchmark datasets for network intrusion detection system. Asian J Res Comput Sci. 2021;7:14–33.
[24] Krishna KV, Swathi K, Rao PRK, Rao BB. A Detailed Analysis of the CIDDS-001 and CICIDS-2017 Datasets. In Pervasive Computing and Social Networking: Proceedings of ICPCSN 2021. Singapore: Springer; 2022. p. 619–38.
[25] Abdul-Jabbar SS, Farhan AK, Luchinin AS. A comparative study of Anemia classification algorithms for international and newly CBC datasets. Int J Online Biomed Eng. 2023;19(6).
[26] Alturfi SM, Muhsen DK, Mohammed MA, Aziz IT, Aljshamee M. A combination techniques of intrusion prevention and detection for cloud computing. In Proceedings of the Journal of Physics: Conference Series. Vol. 1804. Baghdad, Iraq; February 2021. p. 012121.
[27] Hussein AY, Falcarin P, Sadiq AT. Enhancement performance of random forest algorithm via one hot encoding for IoT IDS. Period Eng Nat Sci. 2021;9:579–91.
[28] Alshaikhli S, Farhan AK. A survey on fruit fly optimization algorithm (FOA) in robust secure color image watermarking. In Proceedings of the 2022 Fifth College of Science International Conference of Recent Trends in Information Technology (CSCTIT). Baghdad, Iraq; November 2022. p. 36–42.
[29] Abdul-Jabbar SS. Data analytics and techniques. Aro-The Sci J Koya Univ. 2022;10(2):45–55.
[30] Al-zubidi AF, Farhan AK, Alsadoon A, Khafaga DS, Alharbi AH, El-Kenawy EM. Assessing the effectiveness of techniques in predicting cyberattacks: A DTCF taxonomy. IEEE Access. 2023.
[31] Muhsen DK, Ali SM, Zaki RM, Ahmed AA. Arguments extraction for e-health services based on text mining tools. Period Eng Nat Sci. 2021;9:309–16.
[32] Abd DH, Sadiq AT, Abbas AR. Political Arabic articles classification based on machine learning and hybrid vector. In Proceedings of the 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA). Baghdad, Iraq; November 2020. p. 1–7.
[33] Farhan DAK, Fakhir MR. Forecasting the exchange rates of the Iraqi Dinar against the US dollar using the time series model (ARIMA). Int J Eng Manag Res. 2019.
[34] Fadhil MS, Farhan AK, Fadhil MN. A lightweight aes algorithm implementation for secure iot environment. Iraqi J Sci. 2021;62(9):2759–70. doi: 10/24996/ijs/2021/62/9/25.
[35] Mohammed AA, Al-Ghrairi AHT, Al-zubidi AF, Saeed HM. Unsupervised classification and analysis of Istanbul-Turkey satellite image utilizing the remote sensing. In AIP Conference Proceedings. Vol. 2457. Issue 1. AIP Publishing; 2023, February.
[36] Saini N, Bhat Kasaragod V, Prakasha K, Das AK. A hybrid ensemble machine learning model for detecting APT attacks based on network behavior anomaly detection. Concurrency Comput Pract Exp. 2023;35(28):e7865. doi: 10/1002/cpe/7865.
[37] Haggag M, Tantawy MM, El-Soudani MM. Implementing a deep learning model for intrusion detection on apache spark platform. IEEE Access. 2020;8:163660–72. doi: 10/1109/ACCESS/2020/3022117.
[38] Qazi EUH, Faheem MH, Zia T. HDLNIDS: Hybrid deep-learning-based network intrusion detection system. Appl Sci. 2023;13(8):4921. doi: 10/3390/app13084921.
[39] Al-zubidi AF, Farhan AK, El-kenawy EM. Surveying cyber attack datasets: A comprehensive analysis. JSCCA. 2024.
[40] Muhsen DK, Khairi TWA, Alhamza NIA. Machine learning system using modified random forest algorithm. In Intelligent Systems and Networks: Selected Articles from ICISN 2021, Vietnam. Singapore: Springer; 2021. p. 508–15.
[41] Alsaedi EM, Farhan AK. Retrieving encrypted images using convolution neural network and fully homomorphic encryption. Baghdad Sci J. 2023;20:0206.
[42] Poornima R, Elangovan M, Nagarajan G. Network attack classification using LSTM with XGBoost feature selection. J Intell Fuzzy Syst. 2022;43:971–84.
[43] Inayat U, Zia MF, Mahmood S, Khalid HM, Benbouzid M. Learning-based methods for cyber attacks detection in IoT systems: A survey on methods, analysis, and future prospects. Electronics. 2022;11:1502.
[44] Ali YH, Chinnaperumal S, Marappan R, Raju SK, Sadiq AT, Farhan AK, et al. Multilayered nonlocal bayes model for lung cancer early diagnosis prediction with the internet of medical things. Bioengineering. 2023;10:138.
[45] Khairi TW, Al-zubidi AF, Ahmed EQ. Modified multipath routing protocol applied On Ns3 dcell network simulation system. Int J Interact Mob Technol. 2021;15(10):208.
[46] Muhsen AR, Jumaa GG, AL Bakri NF, Sadiq AT. Feature selection strategy for network intrusion detection system (NIDS) using meerkat clan algorithm. Int J Interact Mob Technol. 2021;15:158.
[47] AL-Bakri NF, Yonan JF, Sadiq AT. Tourism companies assessment via social media using sentiment analysis. Baghdad Sci J. 2022;19:0422.
[48] AL-Bakri NF, Al-zubidi AF, Alnajjar AB, Qahtan E. Multi label restaurant classification using support vector machine. Period Eng Nat Sci. 2021;9:774–83.

[49] Najeeb RF, Dhannoon BN. Classification for intrusion detection with different feature selection methods: a survey (2014–2016). Int J Adv Res Comput Sci Softw Eng. 2017;7:305–11.
[50] Ali YH, Chooralil VS, Balasubramanian K, Manyam RR, Raju SK, Sadiq AT, et al. Optimization system based on convolutional neural network and internet of medical things for early diagnosis of lung cancer. Bioengineering. 2023;10:320.
[51] Alzahrani MY, Bamhdi AM. Hybrid deep-learning model to detect botnet attacks over internet of things environments. Soft Comput. 2022;26(16):7721–35. doi: 10.1007/s00500-021-06132-9.
[52] Jabber SA, Jafer SH. A novel approach to intrusion-detectionsystem: combining lstm and the snakealgorithm. Jordanian J Comput Inf Technol. 2023;9(4).
[53] Idhammad M, Afdel K, Belouch M. Distributed intrusion detection system for cloud environments based on data mining techniques. Procedia Comput Sci. 2018;127:35–41. doi: 10.1016/j.procs.2018.01.006.
[54] Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J Inf Secur Appl. 2020;50:102419. doi: 10.1016/j.jisa.2019.102419.
[55] D'hooge L, Wauters T, Volckaert B, De Turck F. Interdataset generalization strength of supervised machine learning methods for intrusion detection. J Inf Secur Appl. 2020;54:102564. doi: 10/1016/j.jisa/2020/102564.
[56] Kshirsagar D, Kumar S. An ontology approach for proactive detection of HTTP flood DoS attack. Int J Syst Assur Eng Manag. 2021;14:840–7. doi: 10/1007/s13198-021-01167-4.
[57] Khan MA, Kim Y. Deep learning-based hybrid intelligent intrusion detection system. Comput Mater Continua. 2021;68(1):1–16. doi: 10/32604/cmc/2021/015453.
[58] Kim T, Pak W. Hybrid classification for high-speed and high-accuracy network intrusion detection system. IEEE Access. 2021;9:83806–17. doi: 10/1109/ACCESS/2021/3088478.
[59] Shahin M, Chen FF, Hosseinzadeh A, Zand N. Using machine learning and deep learning algorithms for downtime minimization in manufacturing systems: An early failure detection diagnostic service. Int J Adv Manuf Technol. 2023;128(4).
[60] Oleiwi HW, Mhawi DN, Al-Raweshidy H. A meta-model to predict and detect malicious activities in 6G-structured wireless communication networks. Electronics. 2023;12(3):643. doi: 10/3390/electronics12030643.
[61] Chindove H, Brown D. Adaptive machine learning based network intrusion detection. In Proceedings of the International Conference on Artificial Intelligence and its Applications; 2021, December. p. 1–6.