Journal of Wireless Networks and Communication Systems
Vol. 1, Issue. 1, Jan-Mar 2025, pp. 10~21

1D-CNN-based Real-Time Network Intrusion Detection with Privacy-Preserving for IoT

T.A.Sathish Shankar[1], Kunjumol Vadakattu Shamsudeen[2], Rajendrakumar Ramadass[3]
[1]Lecturer, University of Technology and Applied Sciences, Shinas, Sultanate of Oman
[2]Lecturer, University of Technology and Applied Science, Shinas, Sultanate of Oman
[3]Assisstant Trainer, Electrical Section, Engineering Department, College of Engineering & Technology, University of Technology and Applied Sciences, Shinas, Sultanate of Oman.

Article Info

Article history:

Received Jan 8, 2025
Revised Feb 12, 2025
Accepted Mar 7, 2025

Keywords:

Internet of Things(IoT)
Intrusion detection (ID)
1D-CNN (one dimensional-conventional Neural Network)
Privacy preserving
Network security

ABSTRACT

The Internet of Things, or IoT, is a concept that links every device to the Internet and enables them to collaborate to achieve shared goals, including smart automation in the home. The amount of data produced in the rapidly growing Internet of Things (IoT) space has never been higher. Numerous decision-making processes are accelerated by processing this enormous data to get priceless insights. Intrusion detection systems (IDS) are crucial for safeguarding data and computer resources against external threats in computer networks. Modern IDSs struggle to become more effective and flexible in the face of unexpected threats. The suggested approach presents a real-time, privacy-preserving; 1D-CNN-based network intrusion detection system for the Internet of Things. This paper introduces an Intrusion Detection System (IDS) that utilizes a dataset to enhance the Internet of Things (IoT). First, the input data is preprocessed using data processing. Due to its ability to decrease convergence, the choice of features is included in the suggested model used to improve linearity-based principal components analysis. Intrusion detection is recognized using algorithms with extreme gradient boosting (XGBoost) hyperparameters. 1D CNN is the best and least costly model for real-time IoT security monitoring, and it excels in the recommended approach of feature extraction. With a binary intrusion detection accuracy of 99.77%, 1D CNN again outperformed LSTM, RNN, and MLP, which obtained 98.25%, 94.52%, and 92.2%, respectively. The results, based on feature extraction, demonstrate that 1D CNN is a highly efficient real-time IoT security monitoring technique. Among all models that are used for analysis and development, it offers the best efficiency and dependability. Lastly, the performance of the proposed model was superior to that of the existing models.

Corresponding Author:

T.A.Sathish Shankar,
Lecturer, University of Technology and Applied Sciences,
Shinas, Sultanate of Oman
Email: satish.shankar@utas.edu.om

1. INTRODUCTION

Computers, cellphones, software, gadgets, automobiles, food processors, thermostat controls, and other individually identifiable things are all connected via the Internet of Things (IoT) to process and distribute data without the need for human intervention. It uses the Internet to facilitate communication between various devices and maintain data synchronization. The gadgets may be remotely commanded to execute certain tasks [1]. To achieve smart verification of identity, operation, and administration, the Internet of Things (IoT) links everything to the Internet via information-sensing devices. The Internet of Things is more interested in the continuous developments in wireless communications, radio frequency identification, and low-cost sensors. Privacy protection is crucial, nevertheless, considering how quickly IoT network security and privacy issues are developing [2]. As the Internet of Things (IoT) expands, so do the number and

seriousness of security vulnerabilities and threats associated with IoT devices and systems. One well-known technique for quickly identifying IoT attacks and cyber threats to solve such issues is the use of intrusion detection systems, or IDSs [3].

One essential defense mechanism for spotting various security threats in electronic networks is an intrusion detection system (IDS). IDS fall into two primary categories: Intrusion detection systems come in two varieties: host-based (HBIDS) and network-based (NBIDS). To find the intrusion, HBIDS routinely examines a host device's features, including sensors, disk resources; file systems, program logs, and system logs. Meanwhile, the NBIDS examines every packet and looks for suspicious activity. The system may either prohibit access using the source IP address in the network or raise alerts to trigger the following actions when intrusions are detected [4]. IDSs, or intrusion detection devices, are commonly utilized to identify and stop security risks. An ID often employs tactics that are either based on signatures or anomaly-based. While signature-based interference detection methods have demonstrated encouraging outcomes in detecting well-known assault patterns, they are not very good at spotting new threats.

Intrusion detection systems (IDS) are becoming more functional thanks to artificial intelligence (AI), which improved computational models to identify assaults more precisely while reducing false positives. Network performance, throughput, and functionality have declined as assaults on communication devices in networks have escalated [6]. Firewall protection is the IDS's main objective. Although a firewall protects against malicious Internet assaults, an intrusion detection system (IDS) detects whether someone attempts to bypass the firewall's protections and plans to access any computer in the company. If the security device finds any unwanted activity, it alerts the system administrator. An intrusion detection system (IDS) can be hardware or software that examines and tracks user data or network traffic flow for unusual activity to stop intrusions and network attacks. IoT packet or network traffic analysis in real-time. Intrusion detection systems (IDS) employ two basic techniques to analyze network activity and identify any intrusion: anomaly-focused IDS and signature-driven IDS [7]. CNN has been widely working in text picture, and time series data classification. The 1D CNN-based intrusion detection systems proposed in this paper are computationally cheap and offer a faster classification time. The suggested approach consists of three CNN levels in addition to the max pooling and dropping layers [8].

This study proposes a technique for developing a 1D-CNN model that might be applied to building damage diagnosis. The multiple-layer perceptron decision module and the 1D-CNN feature collection module are part of the generic network architecture that was initially created using this technique. Data from 1D vibration signals may be processed directly by a 1D convolutional neural network (1D-CNN). Even with limited training data, high detection accuracy may be attained [9]. This might be applied to the detection of structural problems. The basic concept is to map the relationship between distinguishing features and the degree of structural deterioration to calculate loss. Large amounts of data from monitoring devices may be safely and efficiently analyzed using 1D-CNN.

2. RELATED WORK

Alsaadi et.al [10] proposed a relationship between 1D and 2D CNNs for software-defined systems' ability to identify network intrusions. In the proposed work, we examine how CNNs may be used for NIDS and evaluate different performance metrics and intrusion detection event time complexity. A 1D-CNN outperforms the 2D-CNN and other tested techniques with a classification accuracy of 99.32%, of the data. On the other hand, 2D-CNN performs better than 1D-CNN since it requires less calculation time. The differences between 1D and 2D CNNs' performance for the same layer structure in terms of complexity time and parameter count.

Hossain et.al [11] introduced Deep Learning-Based IoT System Intrusion Identification: A Trustworthy and Successful Approach. The current study provides a deep learning-based approach for real-time threat warning in IoT networks utilizing state-of-the-art models such as 1D Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNNs), and Multi-Layer Perceptrons (MLPs) to enhance intrusion detection. It makes use of the CIC IoT-DIAD 2024 sample. The CIC IoT-DIAD 2024 sample is used. Flow-based sets of characteristics were used to evaluate and train the proposed models, and hyperparameters were changed to maximize the F1 score, accuracy, and recall. In terms of attack classification and feature gathering, the results demonstrate that 1D CNN is the most successful model for real-time IoT intrusion detection, outperforming other models.

Musleh et.al [12] developed IoT Intrusion Detection System Employing ML Algorithms for Data Analysis. Machine learning (ML) is the best approach for intelligent IDS across domains, including the Internet of Things. Since feature extraction methods are crucial to the detection process, the information from IoT systems that are fed into machine learning models must be quickly and accurately gathered. Several feature extractors were evaluated in this work, including image filters and transfer learning models such as

DenseNet and VGG-16. Various machine learning algorithms, including random forest, K-nearest neighbors, SVM, and other models, were evaluated for the feature-gathering approaches the subject of the study.

Alsumaidaee et.al [13] proposed the optoelectronic corona fault identification to identify 1D-CNN, LSTM, and 1D-CNN-LSTM techniques. Three deep learning techniques—1D-CNN, LSTM, and 1D-CNN-LSTM hybrid—are systematically examined in this work to determine the optimal model for coronal flaws. In both the time and frequency domains, the hybrid 1D-CNN-LSTM model is considered the most suitable because of its exceptional accuracy. This model examines the sound waves produced by the device. The model's performance is seen in both the frequency and temporal domains. Among the well-known methods for detecting corona faults in switchgear, the 1D-CNN-LSTM model fared well and can identify corona faults in both the temporal and frequency domains.

Battah et.al [14] proposed using 1D CNNs to detect anomalies in network traffic findings from explainable AI methods. The suggested techniques categorize network traffic data using a one-dimensional convolutional neural network (1D CNN), replicating the identification of anomalies and trends in a cyber-security system. The test results show excellent precision in classification 99.613%, with a rate of mistake of only 0.386%. In other words, this model does a remarkable job of differentiating between abnormal and normal behavior. There were fewer erroneous classifications because the confusion matrix showed significantly larger counts of genuine positives and negatives. After employing Explainable AI (XAI) techniques such as Local Interpretable Model-agnostic Explanations (LIME), we turned to Shapley Compound explanations, such as Shap_value approaches.

Islam et.al [15] introduced Intrusion Detection in IoT Networks Using Machine Learning. Applications that have profited from the deployment of IoT-based solutions include utility services, healthcare, food production, supply chain management, education, transportation, and traffic tracking. However, node heterogeneity brought up security concerns, which are among the most challenging IoT problems. Security measures like access controls, encryption, and authentication cannot be used to secure IoT devices Many IoT threats have been identified in the research suggested, along with shallow (such as decision tree (DT), random forest (RF), or support vector algorithm (SVM)) and deep (such as deep neural networks (DNN), deeper belief network (DBN), long short-term memory (LSTM), stacked LSTM, and bidirectional LSTM (Bi-LSTM)) based intrusion detection systems (IDS). The NSL-KDD, IoTDevNet, DS2OS, IoTID20, and IoT Botnet datasets are among the five benchmark databases used to measure the functioning of these models.

Liu et.al [16] introduced an authentication method for 1D-CNN and GRU-based synchrophasor measuring devices' information sources. The temporal features concealed in the rate, polarization angle, and amplitude data gathered by a one-dimensional convolutional neural network (1D-CNN) in the proposed research project are used by the gated repeated module (GRU) to determine the information source. The widespread use of synchronphasor measuring devices (SMDs) has simplified real-time monitoring and power system control. Meanwhile, in recent years, data faking has become more prevalent. Therefore, investigating data authentication techniques is crucial to successfully identifying and preventing data spoofing. The suggested approach may be able to identify the data source more accurately in less time, according to comparisons of many techniques carried out at the initial stage in large power systems with numerous SMDs.

3. METHODOLOGY

Based on 1D-CNN, the proposed method provides real-time, privacy-preserving network intrusion detection for the Internet of Things. Using this data gathering, this paper proposes an Intrusion Detection System (IDS) to improve IoT. First, the input data is preprocessed using data processing. Due to its ability to decrease convergence, the choice of features is included in the suggested model used to improve linearity-based principal components analysis. Intrusion detection is recognized using algorithms with extreme gradient boosting (XGBoost) hyperparameters. The best and most affordable model for real-time IoT security monitoring is 1D CNN, which excels in feature extraction.

Architecture of 1D-CNN

Figure 1 shows our suggested 1D-CNN design. In the convolutional element, we apply 32 convolutional filters, 5 kernel sizes, 42 characteristics, and additional steps. The activation function of the convolutional component was a "sigmoid". The 1D-CNN architectures may learn a single feature in the first convolution layer by setting a single filter. From the first convolution layer of the network, we create 32 filters to extract 32 distinct features. A 36 x 32 synapse matrix is the end product of the initial convolution layer. Based on the entire length of the input data matrix, each filter has a weight and a certain core measurement. Often referred to as one-dimensional convolutional neural networks (1D-CNNs), these neural networks process one-dimensional data, such as historical data or occur. The spatial and amplitude domains

are used by 1D-CNNs in this study to extract representative features from parallel and non-corona faults. The filters are used in 1D convolution approaches to achieve this.

A confirmation layer with a rectangle line unit (RELU) as the activation function, filters, Fast Convolution (FC), MaxPooling1D, and convolutional layers comprise the CNN algorithm's structure. To prevent overfitting, the number of groups and loss are utilized. The usage of (RELU) as a CNN activation parameter may result in irregularity. Deep CNN initially appeared for image classification, but they are increasingly applied to video detection and analysis. However, the modeling and classification of 1D sequence data is still relatively new. Considering that both corona and non-corona categories may be thought of as an ongoing modeling endeavor, 1DCNNs are an excellent choice. For applications that operate in real-time, compressed 1D-CNNs are advised of their lesser processing requirements [17]. This approach consists of three primary convolutional blocks that become increasingly complex to account for the capacity to abstract features. The first block consists of 32 filters, 64 filters, and 128 filters. The activation functions are 'ReLU', and the kernel size is 2. Each convolutional layer is followed by sequential normalization to reduce internal covariate changes and stabilize learning. Following the first two segmentation blocks, Maximum pooling layers with a pool limit of two are used to extract primary characteristics and reduce the area's dimensions. In the third step, worldwide average pooling is used to minimize the feature maps to a fixed-size result [18].

The proposed 1D-CNN system was constructed with many layers, each of which performs a distinct learning job.

(1) The input layer. The input information usually a 1D feature pattern, is sent to the data input layer. Sequence_length, 1 act as the starting structure in this case, where "sequence length" is the length of the input sequence and "1" denotes that each character in the sequence has a single characteristic.

(2) Convolutional Layers (Conv1D). To extract local properties from the input sequences, two 1D convolution layers—which form the second half of our model—are essential. After receiving the input sequence, the first layer uses a collection of learnable filters to perform a convolution operation. The filter generates the dot products of the filters and a series window and builds a new feature map each time it runs over the input sequence. Two hyperparameters that require optimization are the width of the filtering system, the overall dimension of a kernel, and the variety of filters. With additional filters, the model can extract a large range of characteristics and recognize patterns in a collection of lengths with varying kernel sizes.

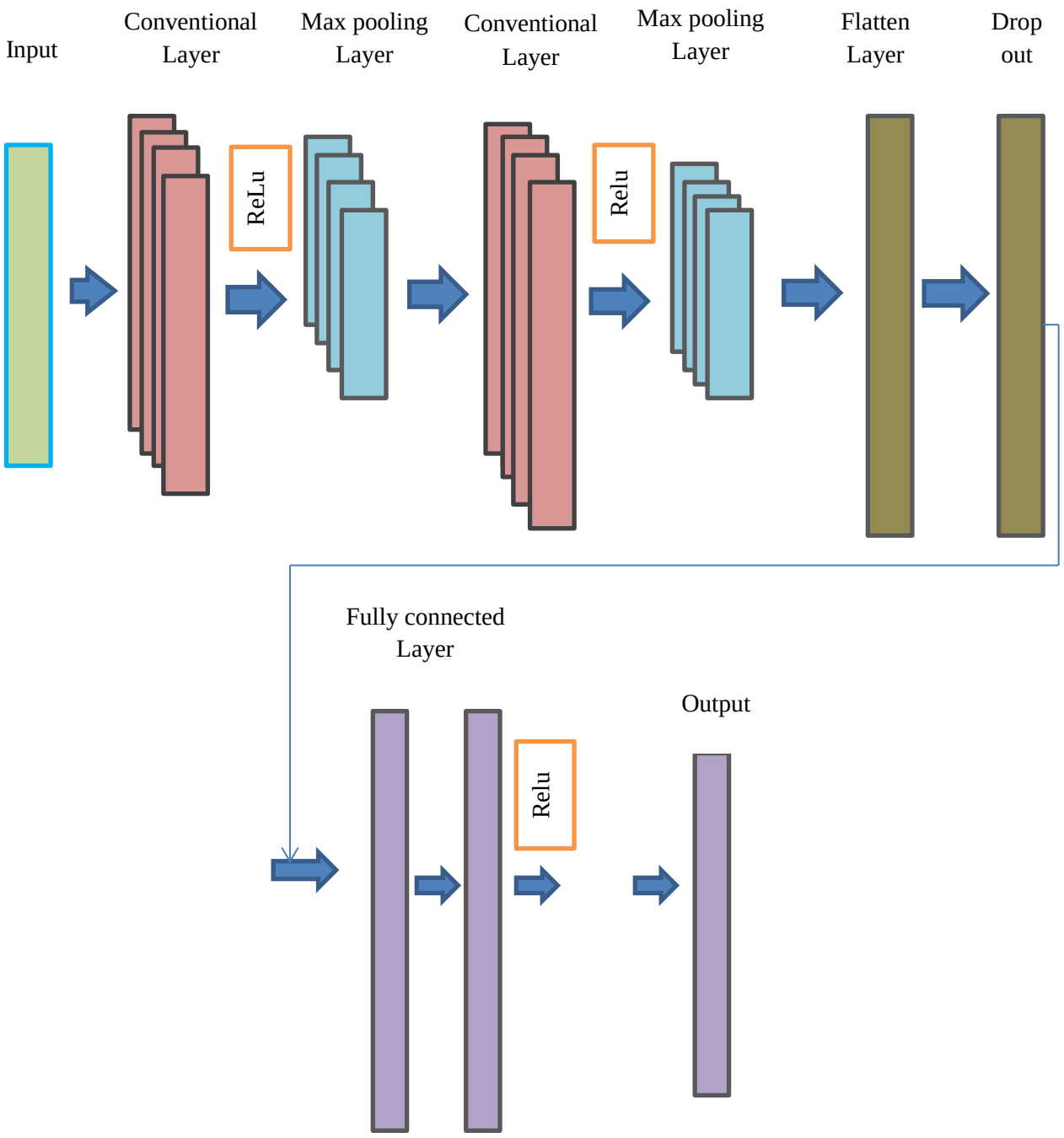

Figure 1. Architecture of 1D-CNN

(3) Max Pooling Layers (MaxPooling1D). A max-pooling layer is used after every convolution layer. The sample reduction of these layers throughout the temporal span of the sequence preserves the most important information while reducing the complexity of the feature maps. The highest value is obtained across a space with a specified combining size using a certain method.

(4) Flatten Layer. A flattened layer is applied after the process has passed through the convolution and pooling stages, converting the two-dimensional feature planning into a one-dimensional vector. Since the convolution and dense layers need input in this format, this layer is essential for linking them.

(5) The output layer. The output layer, which is the final layer in our framework, uses an activation function that is sigmoid to generate an estimated result for the binary categorizing task. The likelihood of the affirmative class may be deduced from this layer's productivity [19].

Dataset

We use NSL-KDD data sets for the proposed model performance evaluation.

Data preprocessing

NSL-KDD is an improvement on the KDDCUP99 dataset. To prevent duplication problems in the previous version, the records in the NSL-KDD have been carefully selected. It just has a moderate amount of records. The NSL-KDD dataset is available in a variety of file formats. KDDTrain+ and KDDTest+ were utilized in this investigation. We can determine that regular traffic makes up 51.88% of the total data, with all forms of assaults accounting for the remaining 48.20%. We separate it into three categories: 70% for teaching, 20% for testing, and 10% for validation. Additional information on NSL-KDD is available. One of the often-used datasets in NIDS is this one [20]. The model's performance was analyzed using the NSL-KDD dataset. The model's output was examined in light of this. The test dataset comprises 63, 913, 3646, 13337, and 19170 records of U2R and R2L, whereas the examples chosen at random for the training phase are 189, 2836, 10431, 40047, and 57883 [21]. Many researchers have validated Network (NIDS) system using the NSL-KDD dataset. Other aggressive tactics are used under these four categories. For example, content elements retain the payment product, time-dependent features evaluate the traffic input over two seconds host-specific features analyze the activity across several established connections, and vital features extract the appropriate data from the communication headers [22].

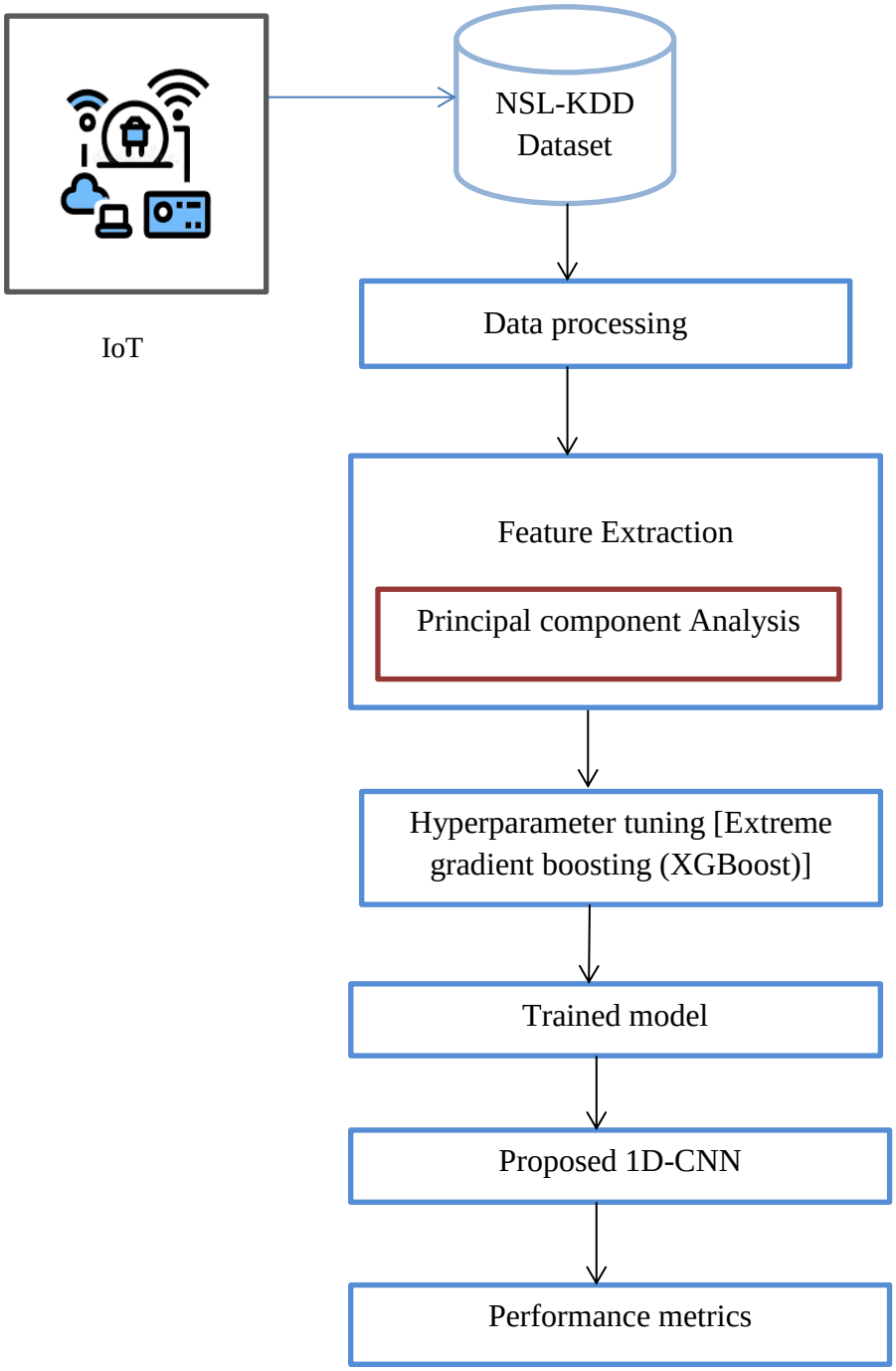

Figure 2. The general procedure of the suggested approach

The broad suggested model technique is depicted in Figure 2. The data set provided by the NSL-KDD is the primary focus of this investigation. After loading the data, it undergoes a pre-processing process. Following pre-processing, the improved linearity-focused gray swarm algorithm chooses features. Features are accurately identified without omitting any information by using principal components analysis (PCA) with enhanced linearity. To facilitate training and testing, the selected data is separated into sets. Hyper-parameter optimization using gradient-based optimization, 1D-CNN, LSTM, RNN, and MLP are used to train the data. The results demonstrate that the optimum model for real-time IoT security monitoring is 1D CNN, which focuses on feature extraction. It offers the best efficiency and dependability among all the research and improvement methods. To integrate the training data, the framework is then modified using the binary classification approach. Measures of the model's performance are evaluated after data training.

Feature Selection with PCA

A key component of machine learning is feature selection. It involves determining the optimum part sequence in the part mix and selecting the greatest attributes from a dataset. A mathematical model has been developed to concurrently reduce the wide goals of setup time and cost [23]. 'An estimating technique called principal component assessment shows the variance form of a collection of data by using many independent components that are proportionate mixes of the beginning variables. The main element is a novel variable that is used to condense vast volumes of associated data into little, uncorrelated findings. Its main components, which are derived from the covariance or relationship matrix (∑), are used for interpretation [24]. The following equation yields the eigenvalues βi and eigenvectors γi, β1, β2,…, βp, which may be computed using both matrices:

$$|\sum -\beta 1| = 0 \quad (1)$$

Meanwhile, the resulting equation yields an eigenvalue γ1, γ2… γp.

$$(\sum - \beta_i I)\gamma_i = 0 \quad (2)$$

Where i = 0, 1, 2,…, p

For instance, if Y1, Y2,…, Yp are random factors distributed in a specific way and have two mutually orthonormal eigenvectors and eigenvalues, (β1, γ1), (β2, γ2), …, (βp, γp), Consequently, the following definition of the ith primary element might be used:

$$\text{PCi} = \gamma i1'Y1 + \gamma i2'Y2 + \cdots + \gamma ip'Yp \quad (3)$$

The first principle component variance is determined as follows using the specification given above:

$$\alpha^2_{PC1} = \beta_i = \gamma'_1 \sum \gamma_1 = \sum_{i=1}^{p} \sum_{i=j}^{p} \gamma_{1i}\, \gamma_{1j}\, \alpha_{ij}; j = 1,2, \ldots, p \quad (4)$$

According to the Lagrangian formula deduction, β1 is an eigenvector that represents βi, and βi above is an eigenvalue that increases the first principal component's (PC1) variability. Another important component, PC2, is responsible for increasing the value of γ'2∑ γ2 = β2. The PCp, which is the pth principal component, controls how much γ'p may be maximized. The initial value of each kth principal component to the whole parameter varies as follows, even if the series of PC1, PC2, ..., PCp should satisfy the conditions of β1 ≥ β2 ≥ …≥ βp:

$$Proporsion = \frac{\beta k}{\beta 1 + \beta 2 + \cdots + \beta p} \quad (5)$$

In situations where every variable can be measured in the same unit, the covariance matrix is utilized. It is necessary to transform the variables into standard normal forms of the same size. The following formula can be used to convert to standard normal form:

$$zp = \frac{yp - \mu p}{\delta p} \quad (6)$$

Extreme gradient boosting (XGBoost)

Recently, XGBoost has become a prominent gradient-boosting decision tree-based technique across a variety of categories due to its superior performance in other algorithms. An enhanced gradient-descent algorithm that drastically cut down the time required for computation, XGBoost can be utilized for classification and regression. It mitigates the overfitting problem by improving the reduction function ascertained by gradient descent using a normalization parameter [25]. The fundamental idea behind a boosting method is to develop the performance of successively integrating the outputs of weak learners. The XGBoost's primary goal is to reduce the normalized cost function, which is as follows:

$$S(1)(t) = \sum n\ i = 1\ |\big(xi^{\nabla}, x(t-1)i + ft(xi)\big) + \delta(ft) \quad (7)$$

Where t is the number of iterations for minimizing the errors, xi is the measured goal value, ˆxi is the predicted category value, the lowering formula that calculates the discrepancy between measured and anticipated values is denoted by l, and (f t) recommends adding a term for regularization to lessen model complexity and excess fitting.

$$fk = \gamma T + \frac{1}{2}\mu ||w||^2 \quad (8)$$

Where T is the total variety of leaves in the tree, w is the scores of each branch, and γ and μ are regularization parameters. Despite the XGBoost model's obvious benefits over other learning methods, performance requires careful hyperparameter tweaking. Recent research has enhanced seven key hyperparameter settings of the XGBoost method: training rate, the maximum depth, minimum children weight, gamma radiation colsample_by_tree, estimators, and subsample.

1DCNN-based detection model

The 1DCNN-IDS model is trained using a subset of Stage 1 features. A 1DCNN captures localized characteristics by applying learned filters, or "kernels," over the sequence. The model can detect and learn from local dependencies in the data, which makes it resistant to distortions and noise. Faster training is also made possible by 1DCNNs' comparatively parameter-efficient design, which shows the result of CNNs'

shared weight architecture. Additionally, 1DCNNs can learn intricate patterns at different sizes thanks to their hierarchical structure, which enhances their versatility and strength in a range of sequential data-related applications [26]. CNN layers are adapted into 1D and used as Cov1D layers. Each Cov1D layer has a kernel size of three, 40 features, and 64 convolution filters. These layers receive "RELU" triggered work, which produces nothing else and, if the input is good, transfers it right away. Next, a dropout layer is added to enhance the output's regularization. The feature maps are then automatically pooled or simplified by a max pooling layer, which creates a condensed representation of the input's recognized features. For this, 1D max pooling is employed. After processing, the output is sent via a flattening [27]. This layer converts the output matrix into a more vector-friendly format efficient categorization. The last stage of the 1DCNN model incorporates a fully linked layer. Two dense layers are formed when this layer breaks. There are 100 neuron monitors and a "Relu" activation algorithm in the first compact layer. On the other hand, a "Softmax" activation function and two feature detectors are incorporated into the final Dense layer architecture.

4. EXPERIMENTAL RESULT AND ANALYSIS

Performance Matrices

Performance metrics are now included in all machine learning projects. For the given datasets, the prediction models were analyzed using performance measures. Performance metrics such as F1-Score, Quality, Precision, and Remember are assessed for the suggested design.

a) Precision (Pc)

Precision is the ratio of properly categorized cases (TRP) to the overall number of accurately categorized (TRP+FLP), which is the technique's covariance unit. It talks about the capital's stability and consistency. Equation 9 is used to calculate it.

$$\mathrm{Pc} = \frac{\mathrm{TRP}}{\mathrm{FLP+TRP}} \tag{9}$$

Where TRP denotes True-Positive and FLP denotes False-Positive respectively

b) Recall (Rc)

In the production measure, Recall is the part that determines the entire number of all optimistic categories that are correct positive categories. The technique used for calculating equation 10 is as follows.

$$\mathrm{Rc} = \frac{\mathrm{TRP}}{\mathrm{FLN+TRP}} \tag{10}$$

Where FLN denotes False-Negative

c) F1-Score

The F1 Score, which can be computed using Equation 11, is the balanced harmonic-mean value of recollection and efficiency.

$$\mathrm{F1-Score} = 2 \times \left(\frac{\mathrm{Rc} \times \mathrm{Pc}}{\mathrm{Rc} + \mathrm{Pc}}\right) \tag{11}$$

d) Accuracy

Accuracy is the system organization rate expressed as the ratio of properly categorized cases (TRN+TRP) to all belongings in the dataset (TRP+FLP+TRN+FLN). Equation 12 is used to compute the accuracy range.

$$\mathrm{Accuracy} = \frac{\mathrm{TRN + TRP}}{(\mathrm{TRP + FLP + TRN + FLN})} \tag{12}$$

Where TRN refers to True-Negative and FLN refers to False-Negative.

Performance Analysis

In this part, we evaluated the model-based suggested approach using several measures, including recall, accuracy, precision, and F1-score. The classifier's performance may be measured by the number of reliable forecasts it produced throughout the sample. Accuracy, precision, recall, and F1 scores of the 1D-CNN, LSTM, RNN, and MLP networks are compared in Table 1. The outcomes also show that the suggested paradigm performs better than alternative strategies already in use.

Table 1. Comparative outcome of 1D-CNN with other existing approaches

Models	Accuracy	Precision	Recall	F1-score
MLP	92.2	94.14	93.34	95.89
RNN	94.52	95.56	90.77	96.6
LSTM	98.25	98.4	98.96	98.32
1D-CNN	99.77	99.08	99.23	99.53

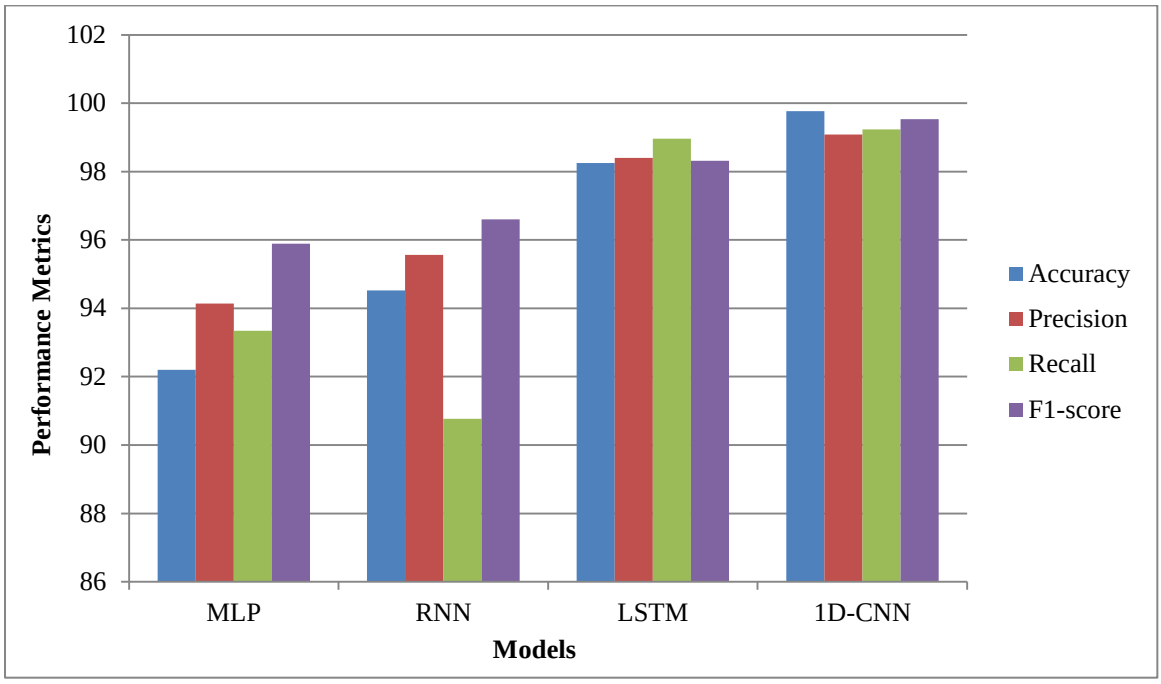

Figure 3. Performance of 1D-CNN with other existing approaches

1D-CNN and the models for intrusion detection systems, LSTM, RNN, and MLP, are compared in Figure 3. With an accuracy of 99.77%, 1D-CNN beat all other models, outperforming RNN (94.52%), LSTM (98.25%), and MLP (92.2%). It also has a strong capacity to reduce false positives and negatives, with a recall rate of 98.2% and an accuracy of 98.6%. SVM showed poor recall of 70% but intermediate accuracy of 80%. KNN produced an outstanding precision of 95.56% but an inferior recall of 90.77%, while MLP lagged with a mediocre recall of 95.89%. Furthermore, 1D-CNN had the best F1-Score of 99.53%, indicating its robustness and balanced performance, outperforming RNN by 96.6%, LSTM by 98.32%, and MLP by 95.89% across all parameters.

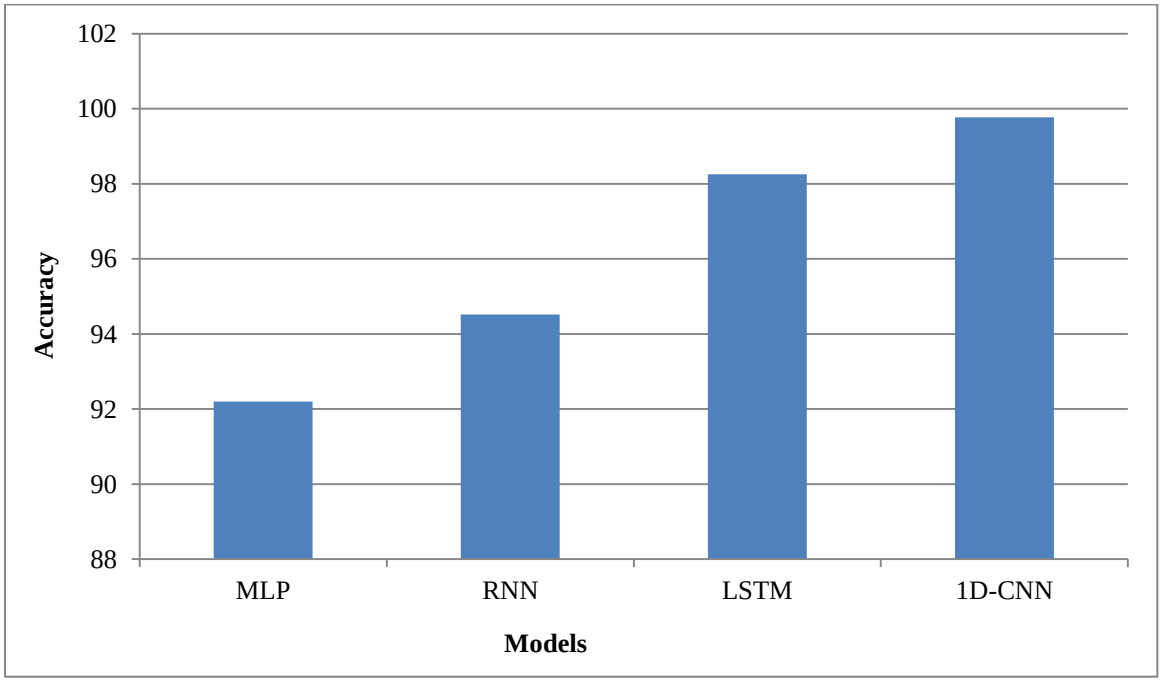

Figure 4. Performance evaluation of the 1D-CNN model using several methods

Figure 4 illustrates a performance analysis of the 1D-CNN method and various current methodologies. The efficiency of the suggested 1D-CNN approach in protecting cloud data privacy is

demonstrated by its achievement of the maximum security level of 99.77%. Furthermore, the security levels attained by the MLP, RNN, and LSTM approaches were 92.2%, 94.52%, and 98.25% accuracy in line with it. The findings show that the suggested PSO technique outperforms current models in attack detection, with the greatest accuracy of 99.77%.

Table 2. Accuracy of 1D-CNN with other existing approaches

Models	Accuracy
MLP	92.2
RNN	94.52
LSTM	98.25
1D-CNN	99.77

5. CONCLUSION

In this paper, we use a deep learning approach and an array of time-series information to create a trustworthy intrusion detection classifier. Given the amount of time required for 1D-CNN training, which we employ for network topologies, the maximum number of batches was set at 32 for the entire network but 64 for merged networks. This research suggested a new 1D-CNN approach for cyberattack classification using IIoT data. First, the input data is preprocessed using data processing. Due to its propensity to reduce convergence, the proposed model, which employs simplified linearity-based basic component evaluation, incorporates choosing characteristics. Intrusion detection is recognized using algorithms with extreme gradient boosting (XGBoost) hyperparameters. With a binary intrusion detection accuracy of 99.77%, 1D CNN once again outperformed LSTM, RNN, and MLP, which obtained 98.25%, 94.52%, and 92.2%, respectively.

REFERENCES

[1] Iftikhar, Z., Javed, Y., Zaidi, S. Y. A., Shah, M. A., Iqbal Khan, Z., Mussadiq, S., & Abbasi, K. (2021). Privacy preservation in resource-constrained IoT devices using blockchain—A survey. Electronics, 10(14), 1732.

[2] Shen, Y., Shen, S., Li, Q., Zhou, H., Wu, Z., & Qu, Y. (2023). Evolutionary privacy-preserving learning strategies for edge-based IoT data sharing schemes. Digital Communications and Networks, 9(4), 906-919.

[3] Ruzafa-Alcázar, P., Fernández-Saura, P., Mármol-Campos, E., González-Vidal, A., Hernández-Ramos, J. L., Bernal-Bernabe, J., & Skarmeta, A. F. (2021). Intrusion detection based on privacy-preserving federated learning for the industrial IoT. IEEE Transactions on Industrial Informatics, 19(2), 1145-1154.

[4] Azizjon, M., Jumabek, A., & Kim, W. (2020, February). 1D CNN-based network intrusion detection with normalization on imbalanced data. In 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 218-224). IEEE.

[5] Umair, M., Tan, W. H., & Foo, Y. L. (2024, September). Optimized 1D Convolutional Neural Network for Efficient Intrusion Detection in IoT Networks. In 2024 IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA) (pp. 1-6). IEEE.

[6] Yang, Y. M., Chang, K. C., & Luo, J. N. (2025). Hybrid Neural Network-Based Intrusion Detection System: Leveraging LightGBM and MobileNetV2 for IoT Security. Symmetry, 17(3), 314.

[7] Sadhwani, S., Khan, M. A. H., Muthalagu, R., & Pawar, P. M. (2024). BiLSTM-CNN Hybrid Intrusion Detection System for IoT Application.

[8] Arsalan, M., Mubeen, M., Bilal, M., & Abbasi, S. F. (2024, August). 1D-CNN-IDS: 1D CNN-based intrusion detection system for IIoT. In 2024 29th International Conference on Automation and Computing (ICAC) (pp. 1-4). IEEE.

[9] Li, X., Guo, H., Xu, L., & Xing, Z. (2023). Bayesian-based hyperparameter optimization of 1D-CNN for structural anomaly detection. Sensors, 23(11), 5058.

[10] Alsaadi, S., Anande, T. J., & Leeson, M. S. (2024, February). Comparative Analysis of 1D-CNN and 2D-CNN for Network Intrusion Detection in Software Defined Networks. In International Conference on Emerging Internet, Data & Web Technologies (pp. 480-491). Cham: Springer Nature Switzerland.

[11] Hossain, M. A. (2025). Deep Learning-Based Intrusion Detection for IoT Networks: A Scalable and Efficient Approach.

[12] Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R. M. (2023). Intrusion detection system using feature extraction with machine learning algorithms in IoT. Journal of Sensor and Actuator Networks, 12(2), 29.

[13] Alsumaidaee, Y. A. M., Paw, J. K. S., Yaw, C. T., Tiong, S. K., Chen, C. P., Yusaf, T., ... & Abdalla, A. N. (2023). Fault detection for medium voltage switchgear using a deep learning hybrid 1D-CNN-LSTM model. IEEE Access, 11, 97574-97589.

[14] Battah, M. H., & AL-Saedi, K. H. (2025, April). Anomaly detection in network traffic using 1D CNNs: Insights from explainable AI techniques. In AIP Conference Proceedings (Vol. 3282, No. 1, p. 020008). AIP Publishing LLC.

[15] Islam, N., Farhin, F., Sultana, I., Kaiser, M. S., Rahman, M. S., Mahmud, M., & Cho, G. H. (2021). Towards machine learning-based intrusion detection in IoT networks. Comput. Mater. Contin, 69(2), 1801-1821.

[16] Liu, S., You, S., Zeng, C., Yin, H., Lin, Z., Dong, Y., ... & Liu, Y. (2021). Data source authentication of synchrophasor measurement devices based on 1D-CNN and GRU. Electric Power Systems Research, 196, 107207.

[17] Mohammed Alsumaidaee, Y. A., Yaw, C. T., Koh, S. P., Tiong, S. K., Chen, C. P., Yusaf, T., ... & Raj, A. A. (2023). Detection of Corona Faults in Switchgear by Using 1D-CNN, LSTM, and 1D-CNN-LSTM Methods. Sensors, 23(6), 3108.
[18] Hassan, B. M., Alomari, E. S., Alrubaye, J. S., & Hassen, O. A. (2025). Adversarially Robust 1D-CNN for Malicious Traffic Detection in Network Security Applications. Journal of Cybersecurity & Information Management, 16(1).
[19] Kilichev, D., & Kim, W. (2023). Hyperparameter optimization for 1D-CNN-based network intrusion detection using GA and PSO. Mathematics, 11(17), 3724.
[20] Hooshmand, M. K., & Huchaiah, M. D. (2022). Network intrusion detection with 1D convolutional neural networks. Digital Technologies Research and Applications, 1(2), 66-75.
[21] Masoodi, F., Bamhdi, A. M., & Teli, T. A. (2021). Machine learning for classification analysis of intrusion detection on NSL-KDD dataset. Turkish Journal of Computer and Mathematics Education, 12(10), 2286-2293.
[22] Liu, J., Kantarci, B., & Adams, C. (2020, July). Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset. In Proceedings of the 2nd ACM workshop on wireless security and machine learning (pp. 25-30).
[23] Sadaram, G., Sakuru, M., Karaka, L. M., Reddy, M. S., Bodepudi, V., Boppana, S. B., & Maka, S. R. (2022). Internet of Things (IoT) Cybersecurity Enhancement through Artificial Intelligence: A Study on Intrusion Detection Systems. Universal Library of Engineering Technology, 9, 1-11.
[24] Surendro, K., Rachmatullah, M. I. C., & Santoso, J. (2022). Improving 1d Convolutional Neural Network (1d Cnn) Performance in Processing Tabular Datasets Using Principal Component Analysis.
[25] Altbawi, S. M. A., Khalid, S. B. A., Mokhtar, A. S. B., Shareef, H., Husain, N., Yahya, A., ... & Alsisi, R. H. (2023). An improved gradient-based optimization algorithm for solving complex optimization problems. Processes, 11(2), 498.
[26] Almi'ani, N., Anbar, M., Karuppayah, S., Sanjalawe, Y., Alrababah, H., Zwayed, F. A., & Hasbullah, I. H. (2024). Feature Selection and 1DCNN-based DDOS Detection in Software-Defined Networking. Engineering Letters, 32(7).
[27] Eren, L., Ince, T., & Kiranyaz, S. (2019). A generic intelligent bearing fault diagnosis system using a compact adaptive 1D CNN classifier. Journal of signal processing methods, 91(2), 179-189.