



ABCNN-IDS: Attention-Based Convolutional Neural Network for Intrusion Detection in IoT Networks

Asadullah Momand¹ · Sana Ullah Jan² · Naeem Ramzan¹

Accepted: 25 May 2024 / Published online: 3 July 2024
© The Author(s) 2024

Abstract

This paper proposes an attention-based convolutional neural network (ABCNN) for intrusion detection in the Internet of Things (IoT). The proposed ABCNN employs an attention mechanism that aids in the learning process for low-instance classes. On the other hand, the Convolutional Neural Network (CNN) employed in the ABCNN framework converges toward the most important parameters and effectively detects malicious activities. Furthermore, the mutual information technique is employed during the pre-processing stage to filter out the most significant features from the datasets, thereby improving the effectiveness of the ABCN model. To assess the effectiveness of the ABCNN approach, we utilized the Edge-IoTset, IoTID20, ToN_IoT, and CIC-IDS2017 datasets. The performance of the proposed architecture was assessed using various evaluation metrics, such as precision, recall, F1-score, and accuracy. Additionally, the performance of the proposed model was compared to multiple ML and DL methods to evaluate its effectiveness. The proposed model exhibited impressive performance on all the utilized datasets, achieving an average accuracy of 99.81%. Furthermore, it demonstrated excellent scores for other evaluation metrics, including 98.02% precision, 98.18% recall, and 98.08% F1-score, which outperformed other ML and DL models.

Keywords Attention mechanism · Convolution neural network · Deep learning · Internet of things · Intrusion detection

1 Introduction

The Internet of Things (IoT) envisions a connected network of various intelligent objects in our surroundings, capable of gathering, processing, and transmitting information [1]. In recent years, the IoT had a significant impact on various industries, including agriculture, medicine, transportation, automobiles, and water monitoring [2–5]. This is the era where

✉ Naeem Ramzan
naeem.ramzan@uws.ac.uk

¹ School of Computing Engineering and Physical Sciences, University of the West of Scotland, Paisley PA1 2BE, UK

² School of Computing Engineering and the Built Environment, Edinburgh Napier University Merchiston Campus, Edinburgh EH10 5DT, UK

all businesses rely on technology, everything is going digital, and as we see, the demand for IoT devices has increased significantly, escalating from 15.42 billion in 2015 to a staggering 35.8 billion in 2021 [6–8]. IoT devices often have limited computational power and memory, making it challenging to implement robust security measures [9, 10]. As businesses deploy more IoT devices, the risk of vulnerabilities being targeted and exploited increases [11]. As shown in Fig. 1, By the year 2025, it is projected that the IoT will reach a staggering number of 75.44 billion devices, resulting in an enormous data output of 79 zettabytes [12]. The IoT has been recognized as a crucial factor in digitization for societal transformation [13, 14].

Many IoT devices gather, save, and handle sensitive data, while their diverse configuration and openness make them an attractive target for attackers [15–17]. Ensuring confidentiality is crucial for the successful implementation of IoT networks. To identify malicious activity, an intrusion detection system (IDS) is necessary to monitor IoT network operations [18–21]. IoT networks often involve a large number of heterogeneous devices, each with its own communication protocols and data formats [22, 23]. Traditional IDS solutions may struggle to handle the diversity and complexity of IoT network traffic, making it difficult to identify abnormal behavior specific to IoT devices [24, 25]. Numerous researchers have collaborated on IDS development, leveraging the power of ML and DL algorithms [26–28]. ML and DL methods find extensive applications in diverse domains including agriculture, medicine, and transportation [29–32]. DL, a subset of ML, is particularly useful for addressing problems involving high-dimensional and intricate data. Moreover, DL enables systematic training of nonlinear models on big datasets.

An imbalanced and inadequate dataset may result in low performance on the current IDS. For instance, consider a security dataset that exhibits imbalanced data, where the disparity between the high and low instances of classes is substantial. The intrusion detection model can be affected by this data imbalance, as it tends to primarily focus on the high-instance classes while disregarding or gradually learning from the low-class instances. As a result, the IoT network utilizing this model may fail to detect attacks that were underrepresented in the training data. Furthermore, a significant challenge in IDS design is feature engineering to extract the most salient attributes. To enhance the effectiveness of existing systems, it is essential to extract the most significant features. To address these issues, this paper proposes an attention-based convolutional neural network (ABCNN) for intrusion

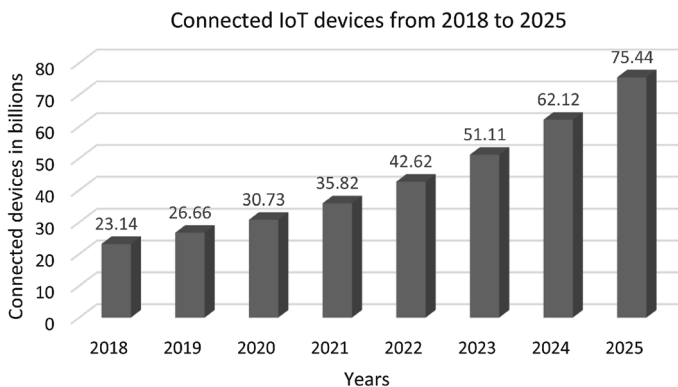


Fig. 1 Projected growth of IoT devices from 2018 to 2025 [12]

detection in IoT networks. The proposed ABCNN employs an attention mechanism that computes attention values for each input attribute. This mechanism aids in the learning process for low-instance classes. On the other hand, the Convolutional Neural Network (CNN) employed in the ABCNN framework converges toward the most important parameters and effectively detects malicious activities [33]. Furthermore, this study utilizes pre-processing techniques such as feature filtering, normalization, and stratified splitting. The mutual information technique is applied during pre-processing to filter out the most significant features from the dataset. The proposed architecture was evaluated using the Edge-IIoTset, IoTID20, ToN_IoT, and CIC-IDS2017 datasets. The performance of the proposed methods was measured using several metrics, including precision, recall, F1-score, and accuracy. The main contributions of this article are:

- In this study, A novel deep learning technique attention-based convolutional neural network (ABCNN) is proposed for intrusion detection in IoT networks. The attention layer computes the attention value for each input, and the CNN is utilized to predict the network's behavior on high-attention features.
- In this study, we employed the mutual information method to select the most significant features. This method calculates the mutual information between each attribute and the target variable based on entropy.
- To demonstrate the effectiveness of the proposed approach in comparison to other several ML and DL methods, a series of experiments were conducted. It is worth noting that all preprocessing steps used in the comparison of the proposed and other models were identical.

The rest of this article is organized as follows: Section 2 presents recent research on intrusion detection in IoT. Section 3 covers the mathematical modeling, overall architecture flow, and experimental methodology. In Section 4, a concise discussion of the experimental results obtained from the proposed model is provided. Finally, Section 5 presents a brief conclusion.

2 Related Work

The proliferation of IoT technology has led to a significant increase in the connectivity of smart devices to the internet. However, this interconnectedness also opens up opportunities for attackers to exploit IoT networks and carry out malicious activities. In response to this pressing issue, numerous researchers have put forth various models aimed at identifying and mitigating such malicious activities in IoT networks.

Altunay et al. [34] proposed a hybrid DL model that incorporates both CNN and long short-term memory (LSTM) for the detection of intrusions in IoT networks. They evaluated the model using the UNSW-NB15 and X-IIoTID datasets for both binary and multi-class classifications. The results section compares the effectiveness of the model with that of CNN and LSTM models and shows that the hybrid CNN and LSTM model outperforms the other models. Wu et al. [35] adopted the Geometric Graph Alignment (GGA) approach to effectively handle the variations in geometry between different domains, thus improving the transfer of intrusion knowledge. In this method, each intrusion domain was represented as a graph, with the vertices and edges corresponding to intrusion categories and their interrelationships. To assess the performance of the GGA approach, the authors employed five publicly available

Table 1 Literature overview

Author	Year	Approach	Dataset	Average accuracy (in %)
Altunay et al. [34]	2023	CNN+LSTM	UNSW-NB15, X-IIoTID	96.35
Wu et al. [35]	2023	GGA	NSL-KDD, UNSW-NB15, CIC-IDS2017, UNSW-BOTIOT, UNSW-TONIOT	71.72
Javadpour et al. [36]	2023	Multi-Agent	KDD Cup 99, NSL-KDD	71.12
Thakkar et al. [37]	2023	DNN based bagging	NSL-KDD, UNSW-NB15, CIC-IDS2017, BOTIOT	98.22
Alghanam et al. [38]	2023	LS-PIO	BoT-IoT, UNSW-NB15, NLS-KDD, KDD Cup 99	96.58
Saba et al. [39]	2022	CNN	NID, BoT-IoT	96.18
Eme et al. [40]	2022	BGH	CIC-IDS-2018, BoT-IoT	99.38
Sharma et al. [41]	2023	DNN	UNSW-NB15	99
El-Ghamry et al. [42]	2023	CNN	NSL-KDD	99

datasets, including NSL-KDD, UNSW-NB15, CIC-IDS2017, UNSW-BoT-IoT, and UNSW-TONIOT. Their proposed model achieved an accuracy of 71.72%, which outperformed other approaches in the comparative analysis of results.

Javadpour et al. [36] introduced a multi-agent-based model designed for the detection and prevention of cyberattacks in the Cloud Internet of Things (CIoT) environment. These agents utilize association rules to effectively identify intrusions. The performance of the multi-agent-based model was assessed using the KDD Cup 99 and NSL-KDD datasets, achieving an accuracy of 71.12%. Thakkar et al. [37] presented a bagging method based on deep neural networks (DNN) to detect intrusions in IoT networks. Their primary emphasis was on addressing the challenge of unbalanced datasets. The evaluation of their proposed bagging model involved the use of NSL-KDD, UNSW-NB15, CIC-IDS2017, and BoT-IoT datasets. Their presented approach yielded an average accuracy of 98.22% across all the datasets. Alghanam et al. [38] introduced the pigeon-inspired optimization local search (LS-PIO) method for the purpose of detecting intrusions in IoT networks. Their proposed method was evaluated using four public datasets, namely BoT-IoT, UNSW-NB15, NLS-KDD, and KDDCUP-99. The LS-PIO method achieved an average accuracy of 96.58% across all the datasets used in the evaluation.

Saba et al. [39] implemented a CNN-based approach for anomaly-based intrusion detection in IoT networks. Their proposed method was trained and evaluated using two distinct datasets: the network intrusion detection (NID) dataset and the Botnet (BoT-IoT) dataset. The CNN model achieved an average accuracy of 96.18% on both datasets. Eme et al. [40] proposed a hybrid model called BGH that utilizes a combination of bi-LSTM and gated recurrent units (GRU) to effectively detect eight known IoT network attacks. The model was trained and evaluated using two widely recognized IoT network traffic datasets: CIC-IDS-2018 and BoT-IoT. Remarkably, the BGH technique achieved an impressive average accuracy of 99.38% on both datasets.

Sharma et al. [41] adopted a deep neural network (DNN) approach for detecting anomalies in IoT networks. They employed a feature filtering technique to extract the most important features from the dataset. To evaluate the performance of their model, they utilized the UNSW-NB15 dataset, achieving 84% accuracy for imbalanced data. However, by utilizing generative adversarial networks (GANs) to balance the data, the accuracy improved significantly to 99%.

El-Ghamry et al. [42] proposed a CNN-based intrusion detection system specifically designed for agriculture IoT networks. They preprocessed the data, selected relevant features, and transformed it into colored images. The authors employed CNN to analyze the images and identify malicious activities within the networks. To evaluate the effectiveness of their system, they used the NSL-KDD dataset, achieving 99% accuracy in their model's outcomes.

A short overview of the literature is presented in Table 1. After reviewing the relevant studies, it becomes clear that numerous studies have focused mainly on a select few classes because of the highly imbalanced datasets. As a result, when dealing with a greater number of attack classes, these systems often encounter difficulties in achieving precise detection. In contrast, this paper presents a novel method known as ABCNN, which improves the effectiveness of current models for both smaller and larger sets of attack classes.

3 The Proposed Attention-Based CNN

This study proposes an attention-based convolutional neural network (ABCNN) model for detecting malicious attacks. The model consists of an attention layer and convolutional neural network (CNN) layers (Fig. 2). The attention layer calculates the attention value of each input attribute/element, while the CNN layers focus on the importance of each attribute and predict the behavior of the network. The basic architecture of a CNN for intrusion detection usually comprises one or more convolutional layers, pooling layers, and fully connected layers [43, 44]. For the proposed model, we utilized one convolutional layer, one max-pooling layer, and three fully connected layers. This decision is based on the results illustrated in Tables 4, 5, 6, and 7. It can be clearly seen from the results in these tables that the model based on selected set of configurations performs the best among all other configurations when applied to all the different datasets (Fig. 2).

The proposed model uses the attention layer to calculate a set of attention weights that indicate the importance of each key element for the current query. This is achieved by computing a similarity score between the query and each key element and then applying a softmax function to obtain a set of normalized attention weights. Once the attention weights have been calculated, they are used to weight the values and summed up to obtain the output of the attention layer. The attention weights a_{ij} for each query i and key j in the sets of queries Q , keys K , and values V are calculated using Eq. (1).

$$a_{ij} = \text{softmax} \left(\frac{Q_i K_j^T}{\sqrt{d_k}} \right) \quad (1)$$

Here, Q_i represents the i -th query, K_j represents the j -th key, d_k represents the dimensionality of the key vectors, and softmax refers to the softmax function used for computing the attention weights. Next, the attention weights are used to weight the corresponding values, and the resulting weighted values are summed up to obtain the output (X) of the attention layer. This process is described mathematically by Eq. (2).

$$X_i = \sum_j a_{ij} V_j \quad (2)$$

Here, X_i represents the i -th output (X), V_j represents the j -th value, and the sum is taken over all keys j . The input to the CNN layer is obtained as the output of the previous layer. Eqs. (3) and (4) provide a demonstration of the 1D convolutional layer.

$$x_u = b_u + \sum_{i=1}^N (s_i, w_{iu}) \tag{3}$$

$$y_u = f(x_u) \tag{4}$$

Where x_u represents the input in 1D convolutional layer, and the output of the previous layer neurons are denoted by s_u . The kernel from i to u is represented by w_{iu} , and the bias value of the neuron in the convolutional layer is denoted by b_u . The ReLU activation function is used in convolutional layers which is represented by $f()$, and its mathematical form is presented in Eq. (5). The output of the 1D convolutional layer is denoted by y_u , which becomes the input to the max pooling layer, as indicated in Eq. (6). During pooling, the maximum values from the output of the convolutional layer within the region \mathfrak{R}

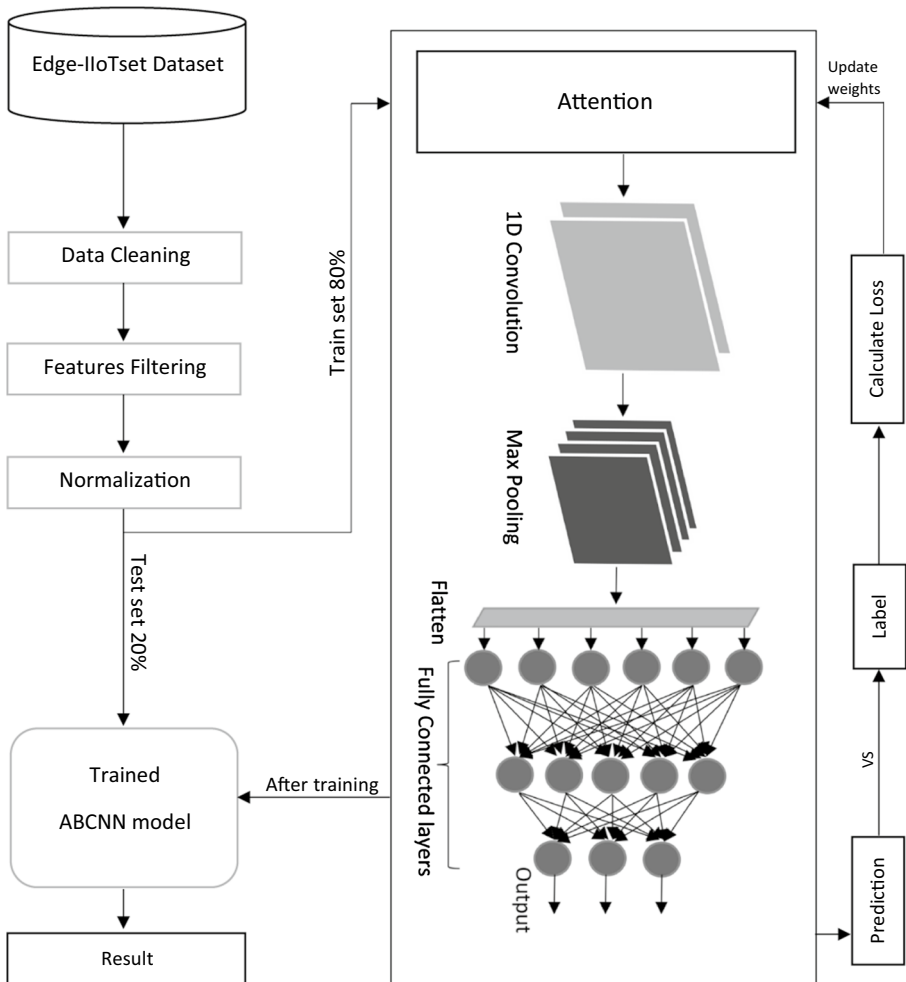


Fig. 2 Proposed ABCNN model architecture

are selected, and the result is denoted by s_u . Once the max-pooling layer is applied, the flatten method is utilized to convert the output shape of the final pooling layer into a one-dimensional array, which is subsequently fed as input to the fully connected layers. In the fully connected layers, the ReLU activation function is employed. Finally, the last fully connected layer utilizes the softmax function, as demonstrated in Eq. (7), to produce the final result. The hyperparameters used in the proposed approach are listed in Table 2.

$$f(x_u) = \max(0, x_u) \quad (5)$$

$$s_u = \max_{i \in \mathcal{R}} y_u \quad (6)$$

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^u e^{x_j}} \quad (7)$$

4 Performance Evaluation

This section provides details of the preliminary measures required to implement the approach, such as data cleaning, feature filtering, and normalization. Furthermore, it presents the test results of the proposed ABCNN architecture for identifying malicious attacks in IoT networks. Various experiments were conducted on the proposed model to identify the optimal configuration. These experiments were conducted with varying model layers, batch sizes, and optimization functions. For the purposes of this specific experiment, a sparse categorical cross-entropy loss function was utilized for loss calculation, and the model was trained over six epochs. To evaluate the effectiveness of the proposed model, a five-fold cross-validation method was employed.

4.1 Datasets

In this study, we utilized Edge-IIoTset, IoTID20, ToN_IoT, and CIC-IDS2017 datasets which are well-established datasets that are frequently used in the area of ML-based IDSs by researchers. The Edge-IIoTset dataset includes IoT and IIoT network communication instances obtained from a real-world testbed implementation of seven interconnected layers, including cloud computing, network functions virtualization, blockchain network, fog computing, software-defined networking, and edge computing, as well as IoT and IIoT perception layers [45]. The data was generated by more than ten different types of devices, such as water and soil measuring, temperature, humidity, and other IoT devices. Edge-IIoTset contains fourteen attacks related to IoT and IIoT communication protocols, which are divided into five categories: DoS/DDoS, Information gathering, Man in the middle, Injection, and Malware attacks. Data was collected from network packets using pcap

Table 2 The Utilized hyperparameters

Optimizer	Learning rate	Loss function	Batch size	Filters	Kernal size	Pooling size
Adam	0.001	Sparse categorical crosentropy	32	64	3	2

files, which were then converted to CSV using the Zeek and TShark tools. The dataset contains 2219201 samples to evaluate the DL methods. The IoTID20 dataset was created specifically for detecting cyberattacks in IoT networks. It was generated by utilizing home-connected smart devices, such as SKT NGU and EZVIZ Wi-Fi cameras [46]. The significant benefit of this dataset lies in its incorporation of contemporary communication data and fresh insights into network interference detection. In total, the dataset encompasses 83 distinct features related to IoT networks [47]. The ToN_IoT dataset was derived from a real-time IoT network at UNSW Canberra in Australia. This compilation encompasses seven distinct categories of cyberattacks targeting IoT networks, each documented within its respective file [48]. The CIC-IDS2017 dataset was generated using real-time network data, spread across eight distinct files. These files encompass a five-day period, capturing both regular and attack-related traffic data [49]. This dataset was created by the Canadian Institute of Cybersecurity [50]. Table 3 provides summaries of the utilized datasets.

4.2 Data Cleaning

This is the very first process in the preprocessing phase, and it addresses the issue of null values and converting categorical attributes to numeric attributes. The Edge-IIoTset dataset contains no missing value. The used dataset has categorical data which include a variety of data categories. To transform categorical data into numeric, first, we considered using a one-hot encoder; however, this technique requires a huge memory and exposes significant latency [51]. As a necessary consequence, for the transformation task, we instead used the label encoder mechanism. The label encoder strategy assigns a unique numeric value to each label based on alphabetical order and does not require any additional memory.

4.3 Features Filtering

Each dataset includes a set of attributes. If a dataset contains insignificant attributes that do not influence the output, it is advisable to remove them from the dataset. These features can lead to overfitting and underfitting problem, affecting the computation time and the effectiveness of the framework. The selection of features is a method for eliminating insignificant attributes from a dataset and retaining only essential attributes. The primary goal of feature filtering is to avoid overfitting and underfitting, enhance efficiency, and decrease the model's training and response times.

In this experiment, we employed the mutual information method to select the most significant features for training the models. Mutual information is a fundamental concept in information theory that quantifies the shared information between two random variables. It measures the extent to which knowing the value of one variable reduces the uncertainty about the other variable. Mutual information is employed to identify relevant features that contribute to the detection of malicious activities. This method calculates the mutual information between each attribute and the target variable based on entropy. Mathematically, the mutual information is expressed in Eq. (8), where $I(X; Y)$ denotes the mutual information between X and Y , $p(x, y)$ represents the joint probability mass function of X and Y , and $p(x)$ and $p(y)$ correspond to the marginal probability mass functions of X and Y , respectively. Only the features with a threshold value greater than 0.1 were selected. The datasets used in this experiment, namely Edge-IIoT set and IoTID20, contain a total of 61

Table 3 The utilized datasets detail presentation for each class

Edge-IIoTset		IoTID20		ToN_IoT		CIC-IDS2017	
Classes	Instances	Classes	Instances	Classes	Instances	Classes	Instance
Fingerprinting	1001	Mirai	415309	Backdoor	35000	DoS	252672
MITM	1214						
Ransomware	10925			Injection	35000	PortScan	158930
XSS	15915	Scan	75265				
Port_Scanning	22564			Password	35000	DDoS	128027
Backdoor	24862						
Uploading	37634	DoS	59391	DDoS	25000	Patator	13835
DDoS_HTTP	49911						
DDoS_TCP	50062			Ransomware	16030	Web Attack	2180
Vulnerability_Scanner	50110	MITM	35377				
Password	50153	ARP		XSS	6116	Bot	1966
SQL_injection	51203	Spoofing					
DDoS_ICMP	116436			Scanning	3973		
DDoS_UDP	121568						
Normal	1615643	Normal	40073	Normal	245000	Benign	529918

and 83 features, respectively. Out of these, 29 features were selected from the Edge-IIoT set, while 55 features were chosen from the IoTID20 dataset.

$$I(X;Y) = \sum \sum p(x, y) \log \left[\frac{p(x, y)}{p(x) \cdot p(y)} \right] \quad (8)$$

4.4 Normalization

In this experiment, the selected datasets have different features which have different range values, for example, some features have very large range values, and some attributes have very small range values. To begin addressing this issue, we used the min-max normalization methods to normalize all attributes between 0 and 1 by using Eq. (9). Where x is the original value to be normalized, x_{min} and x_{max} are the minimum and maximum values of the feature, respectively, and x_{norm} is the normalized value of x . After normalization, split data into 80% train and 20% test sets by using stratified split. A distribution of 80% training and 20% testing split was used to facilitate 5-fold cross-validation. This distribution ensures that each fold of the cross-validation process encompasses 20% for testing and 80% for training, effectively covering the entire dataset for robust performance validation. The stratified method split data equally for each class in the train and test sets.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (9)$$

4.5 Experimental Setup

To conduct the experiment on the proposed model, Python 3.11 was utilized since it is widely used among scientists for experimentation. The Keras package from the TensorFlow library was employed, as it is a convenient deep-learning framework. For the code implementation, Jupyter Notebook was utilized, as it provides results after each code cell is executed. The experiment was conducted on a Windows 10 operating system, utilizing an i5-8th generation laptop with 24 GB RAM and a 1.8 GHz processor. The dataset employed in the experiment was significant, necessitating the use of high RAM.

4.6 Evaluation Measure

In this study, we utilize multiple metrics to evaluate the effectiveness of our proposed classification approach. These metrics include accuracy (ACC), macro precision (MP), macro recall (MR), and F1-score which are based on true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Precision is a technique for evaluation of the model that shows how much correct data are given in total returned data. Precision will find from the testing results of the model which returns a confusion matrix. Precision is calculated from TP and FP values. This experiment uses the MP which is calculated by using Eq. (10).

$$MP = \frac{1}{k} \sum_{i=1}^k \frac{\alpha_i}{\alpha_i + \beta_i} \quad (10)$$

where k is the number of classes, α_i represents the true positives for class i , and β_i represents the false positives for class i . Recall is a technique for evaluation of the model that shows how much correct data are returned from a total collection of data. Recall will find from the testing results of the model which return a confusion matrix. The recall is calculated from TP and FN values. This experiment uses the MR which is calculated by using Eq. (11).

$$MR = \frac{1}{k} \sum_{i=1}^k \frac{\alpha_i}{\alpha_i + \sigma_i} \quad (11)$$

where σ_i represents the false negatives for class i . The F1-score is a technique for evaluation of the model that combines both techniques precision and recall and gives a single value. The F1-score is calculated using Eq. (12).

$$F1\text{-score} = \frac{2 \times MP \times MR}{MP + MR} \quad (12)$$

Accuracy is a metric used to evaluate the performance of a model in accurately detecting attacks. Eq. (13) is employed to calculate accuracy, where γ_i represents the true negatives for class i .

$$Accuracy = \frac{\sum_{i=1}^k \alpha_i}{\sum_{i=1}^k (\alpha_i + \beta_i + \sigma_i + \gamma_i)} \quad (13)$$

4.7 Proposed ABCNN Layers Comparison

The experiments were carried out with varying numbers of convolutional one-dimension (Conv1D) and dense layers, and the results were compared to find the optimal layers. In this experiment, the Adam optimization function with batch size of 32 was employed. Figs. 3, 4, 5, and 6 depicts the training and validation performance, respectively, providing an assessment of the model's performance in terms of overfitting and underfitting. These figures demonstrate that the model is capable of generalizing well to unseen data and does not exhibit any signs of overfitting. The performance evaluation was then recorded in Tables 4, 5, 6, and 7 on all the utilized datasets, which presents the accuracy and other performance metrics for different layers. The results indicate that the best performance was achieved using one Conv1D and three fully connected layers in the proposed ABCNN approach. This means that this particular configuration of layers provides optimum performance for the task at hand. Furthermore, it is stated that all the evaluation metrics were optimal for these layers. This suggests that this configuration consistently performed better than any other number of layers tested, indicating that it is a robust and reliable choice for this task.

4.8 Proposed ABCNN Results on Different Optimization Functions

A series of experiments were conducted to evaluate various optimization functions and determine the most optimal one for the proposed model. The experiments were conducted using the optimized layer configuration of the proposed model, as explained previously, with a batch size of 32. The training accuracy and loss of the proposed model on different optimizers are compared in Figs. 7, 8, 9 and 10. The results of the experiments, comparing the performance of the ABCNN model using different optimization functions, are summarized in Tables 8, 9, 10 and 11 on all the utilized datasets. The findings reveal that the ABCNN model achieved superior performance when the Adam optimization function was employed for both datasets, outperforming the other functions.

4.9 Proposed ABCNN Results on Different Batch Sizes

As previously mentioned, various experiments were conducted to determine the optimal batch size for the proposed model. For this specific experiment, the proposed model's optimal layer configuration was utilized with the Adam optimization function. The results for the proposed ABCNN model using different batch sizes are presented in Tables 12, 13, 14 and 15 on all the utilized datasets. The findings indicate that the ABCNN model performed better with the batch size 32 function than others for both datasets.

4.10 Performance Comparison with Other ML and DL Methods

The proposed ABCNN model was evaluated to determine its effectiveness in comparison with other ML and DL approaches, such as CNN, LSTM, GRU, Naive Bayes (NB), and Support Vector Machines (SVM). All models were evaluated under the same experimental conditions, including the preprocessing steps and the dataset split into 80% training and 20% testing sets. Tables 16, 17, 18, and 19 provide a detailed analysis of the performance evaluation of the proposed ABCNN model in network behavior classification on all the

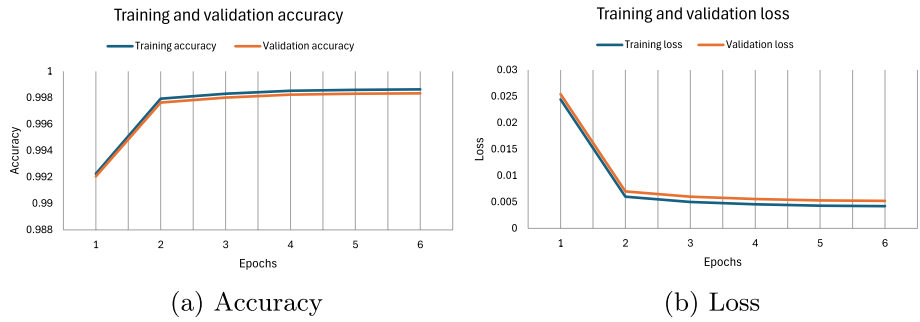


Fig. 3 Training and validation performance of the proposed ABCNN model on Edge-IIoTset dataset

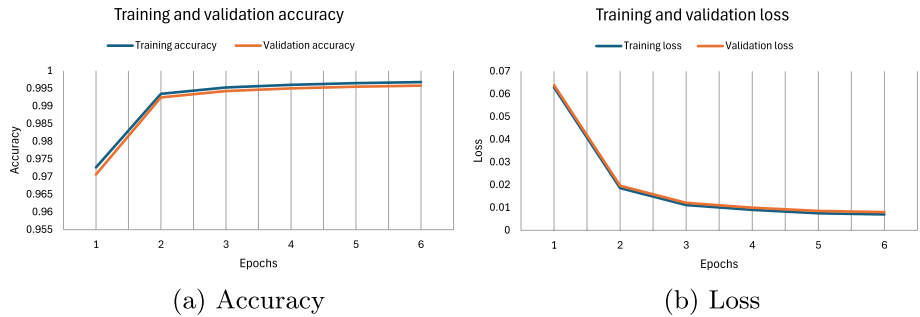


Fig. 4 Training and validation performance of the proposed ABCNN model on IoTID20 dataset

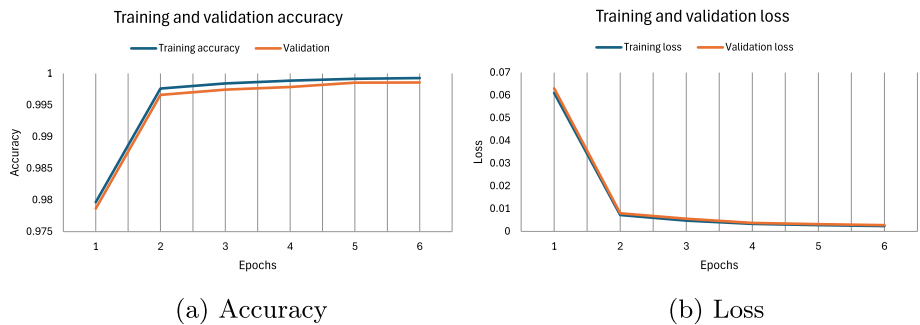


Fig. 5 Training and validation performance of the proposed ABCNN model on ToN_IoT dataset

utilized datasets. The results show that the proposed ABCNN model outperformed all other ML and DL approaches, including CNN, LSTM, GRU, NB, and SVM. Therefore, the proposed ABCNN model can be considered an effective approach for achieving high performance in the given task.

It can be seen in Table 16 that the M-Precision of CNN approach is better than the proposed approach. Similarly, the Table 19 shows that the M-Precision of GRU approach is better than the proposed approach. These high precision and low recall of CNN and GRU suggest that when these models predict a positive class, they are very

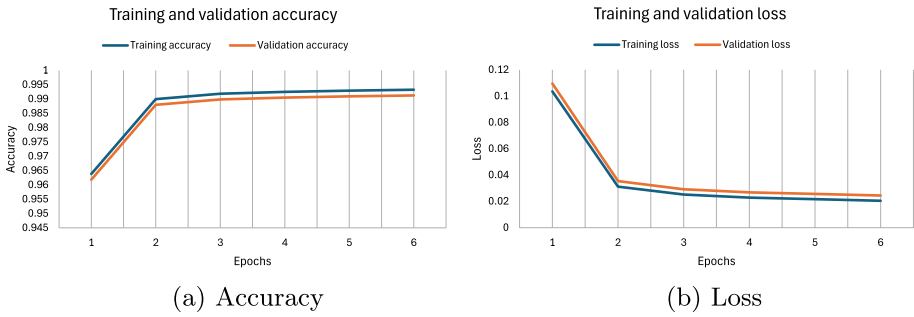


Fig. 6 Training and validation performance of the proposed ABCNN model on CIC-ID2017 dataset

Table 4 Layers performance comparison of proposed ABCNN on Edge-IIoTset dataset

ConvID	Dense	Accuracy	M-Precision	M-Recall	F1-Score
1	1	0.9992	0.996	0.9846	0.9897
1	2	0.9993	0.9938	0.9882	0.9909
1	3	0.9994	0.9921	0.9953	0.9936
2	1	0.9993	0.9918	0.9945	0.9928
2	2	0.9993	0.9904	0.9832	0.9865
2	3	0.9993	0.9857	0.9946	0.9897
3	1	0.9993	0.9842	0.9915	0.9877
3	2	0.9984	0.9881	0.9277	0.9438
3	3	0.9986	0.9888	0.978	0.9824
4	1	0.9992	0.9918	0.9796	0.9849
4	2	0.9982	0.991	0.9384	0.9495
4	3	0.9981	0.9797	0.9415	0.9474

Table 5 Layers performance comparison of proposed ABCNN on IoTID20 dataset

ConvID	Dense	Accuracy	M-Precision	M-Recall	F1-Score
1	1	0.9973	0.9973	0.9964	0.9968
1	2	0.9967	0.996	0.9949	0.9954
1	3	0.9981	0.9973	0.9964	0.9969
2	1	0.9973	0.9971	0.9965	0.9968
2	2	0.9964	0.9962	0.9937	0.9949
2	3	0.9958	0.9933	0.9961	0.9936
3	1	0.9969	0.9962	0.9957	0.9959
3	2	0.996	0.9951	0.9939	0.9945
3	3	0.9959	0.9948	0.9931	0.9939
4	1	0.9964	0.9966	0.9938	0.9942
4	2	0.9956	0.9953	0.9917	0.9935
4	3	0.9944	0.9947	0.9883	0.9915

likely correct (high precision), but they miss a significant number of positive cases (low recall). This could be because CNN and GRU are too conservative in their predictions, possibly due to being too sensitive to certain features that do not generalize well. While

Table 6 Layers performance comparison of proposed ABCNN on ToN_IoT dataset

Conv1D	Dense	Accuracy	M-Precision	M-Recall	F1-Score
1	1	0.9999	0.9995	0.9996	0.9995
1	2	0.9999	0.9997	0.9996	0.9996
1	3	0.9999	0.9999	0.9998	0.9998
2	1	0.9995	0.9992	0.9985	0.9988
2	2	0.9991	0.9989	0.9986	0.9987
2	3	0.9999	0.9998	0.9997	0.9997
3	1	0.9999	0.9998	0.9997	0.9997
3	2	0.9999	0.9998	0.9997	0.9997
3	3	0.9998	0.9996	0.9984	0.9989
4	1	0.9982	0.9975	0.9974	0.9975
4	2	0.9998	0.9997	0.9995	0.9996
4	3	0.9996	0.9994	0.9993	0.9994

Table 7 Layers performance comparison of proposed ABCNN on CIC-ID2017dataset

Conv1D	Dense	Accuracy	M-Precision	M-Recall	F1-Score
1	1	0.9937	0.9536	0.9258	0.9354
1	2	0.9942	0.9918	0.8034	0.8265
1	3	0.9952	0.9318	0.9359	0.9329
2	1	0.9942	0.9914	0.8112	0.8426
2	2	0.9938	0.9231	0.9261	0.9117
2	3	0.9944	0.9314	0.9331	0.9231
3	1	0.9927	0.8684	0.9438	0.8985
3	2	0.9944	0.9766	0.8236	0.8569
3	3	0.9931	0.9822	0.8077	0.8256
4	1	0.9934	0.8881	0.9341	0.9043
4	2	0.9915	0.839	0.7974	0.8108
4	3	0.9901	0.9014	0.9209	0.8937

Table 8 Performance comparison of proposed ABCNN on different optimizers on Edge-IIoTset dataset

Optimizer	Accuracy	M-Precision	M-Recall	F1-Score
Nadam [52]	0.9991	0.9845	0.9941	0.9886
Adamax [52]	0.9989	0.9951	0.9838	0.9891
SGD [52]	0.9974	0.9721	0.9131	0.9226
RMSprop [52]	0.9922	0.8972	0.8693	0.8776
Adadelta [53]	0.9208	0.3914	0.4605	0.4176
Adagrad [54]	0.9711	0.6635	0.6827	0.6712
Adam [55]	0.9994	0.9921	0.9953	0.9936

low precision and high recall of the proposed ABCNN indicate that the model identifies most of the positive cases (high recall) but also makes a lot of false positive errors (low precision). This might happen if the model is generalized and captures many features as indicators of the positive class.

Table 9 Performance comparison of proposed ABCNN on different optimizers on Edge-IoTID20 dataset

Optimizer	Accuracy	M-Precision	M-Recall	F1-Score
Nadam [52]	0.9968	0.9961	0.9951	0.9956
Adamax [52]	0.9964	0.9963	0.9936	0.997
SGD [52]	0.9931	0.9897	0.9881	0.9889
RMSprop [52]	0.9957	0.9951	0.9932	0.9941
Adadelata [53]	0.7543	0.4379	0.4229	0.4202
Adagrad [54]	0.8341	0.8099	0.6836	0.6836
Adam [55]	0.9981	0.9973	0.9964	0.9969

Table 10 Performance comparison of proposed ABCNN on different optimizers on ToN_IoT dataset

Optimizer	Accuracy	M-Precision	M-Recall	F1-Score
Nadam [52]	0.9999	0.9997	0.9998	0.9998
Adamax[52]	0.9999	0.9999	0.9997	0.9998
SGD [52]	0.9974	0.9863	0.9906	0.9884
RMSprop [52]	0.9997	0.9995	0.9983	0.9989
Adadelata [53]	0.7634	0.7622	0.251	0.3776
Adagrad [54]	0.9234	0.6549	0.6763	0.6598
Adam [55]	0.9999	0.9999	0.9998	0.9998

Table 11 Performance comparison of proposed ABCNN on different optimizers on CIC-IDS2017 dataset

Optimizer	Accuracy	M-Precision	M-Recall	F1-Score
Nadam [52]	0.9951	0.9311	0.9335	0.9321
Adamax [52]	0.9932	0.9257	0.9205	0.9105
SGD [52]	0.9400	0.6697	0.6069	0.6246
RMSprop [52]	0.989	0.9228	0.9089	0.9034
Adadelata [53]	0.8014	0.4376	0.4434	0.4326
Adagrad [54]	0.8956	0.5076	0.5155	0.5099
Adam [55]	0.9952	0.9318	0.9359	0.9329

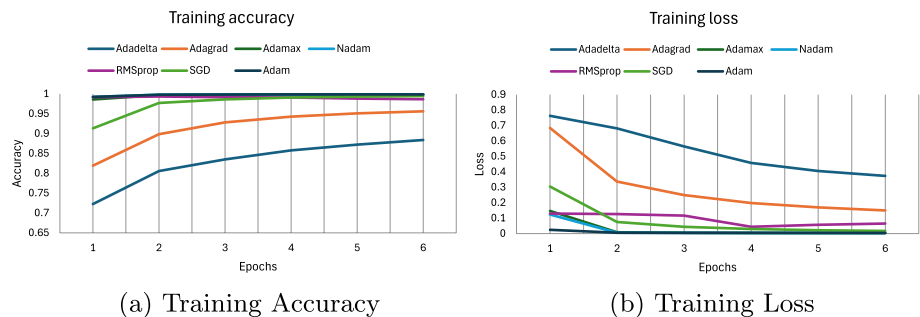


Fig. 7 Training performance of ABCNN on different optimizers using Edge-IIoTset dataset

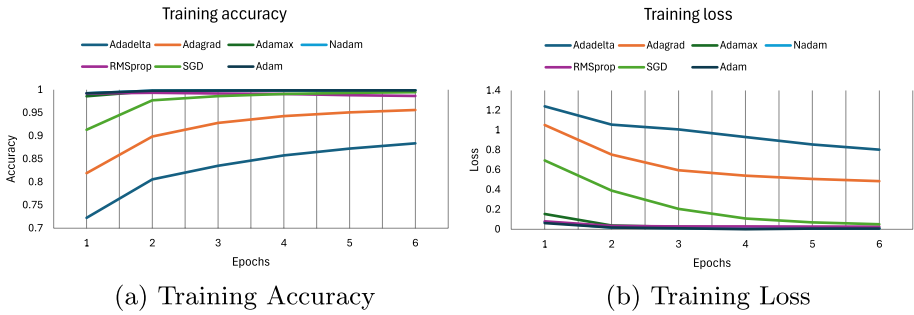


Fig. 8 Training performance of ABCNN on different optimizers using IoTID20 dataset

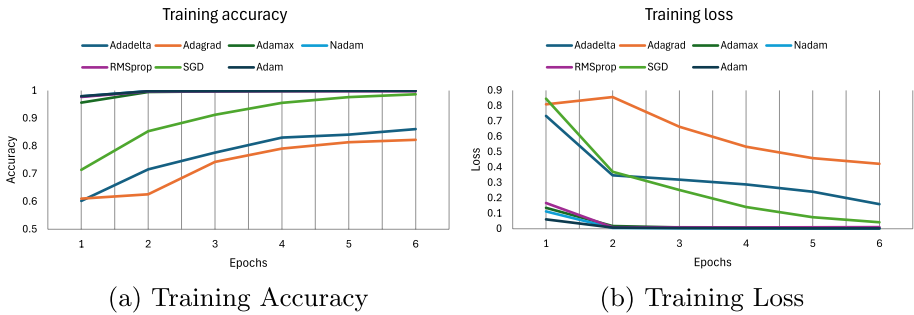


Fig. 9 Training performance of ABCNN on different optimizers using ToN_IoT dataset

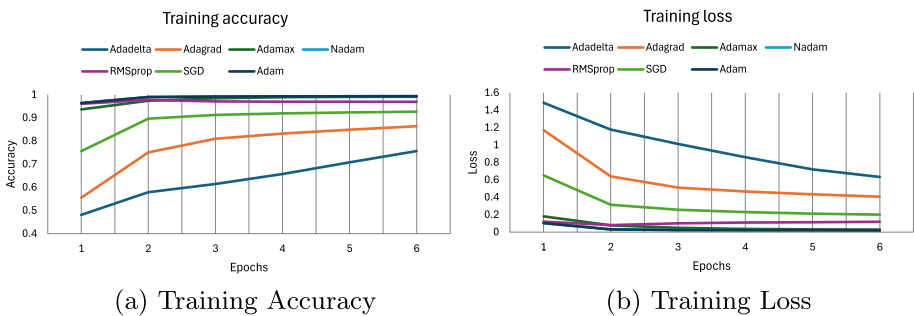


Fig. 10 Training performance of ABCNN on different optimizers using CIC-IDS2017 dataset

5 Conclusion

In this study, we have proposed an attention-based convolutional neural network (ABCNN) for intrusion detection in IoT networks. Moreover, we have utilized the mutual information technique during the pre-processing stage to filter out the most relevant features from the dataset. To evaluate the effectiveness of our proposed ABCNN approach, we have employed the Edge-IoTset, IoTID20, ToN_IoT, and CIC-IDS2017 datasets. The performance of our approach has been compared with other Intrusion Detection Systems (IDS) based on both ML and DL techniques. The results clearly demonstrate that our proposed

Table 12 Performance of proposed ABCNN on different batch sizes using Edge-IIoTset dataset

Batch Size	Accuracy	M-Precision	M-Recall	F1-Score
32	0.9994	0.9921	0.9953	0.9936
64	0.9992	0.9944	0.9891	0.9915
128	0.9993	0.9929	0.9929	0.9929
256	0.9993	0.9945	0.9886	0.9913

Table 13 Performance of proposed ABCNN on different batch sizes using IoTID20 datasets

Batch Size	Accuracy	M-Precision	M-Recall	F1-Score
32	0.9981	0.9973	0.9964	0.9969
64	0.997	0.9958	0.996	0.9959
128	0.9968	0.995	0.9953	0.9931
256	0.9968	0.9966	0.995	0.9958

Table 14 Performance of proposed ABCNN on different batch sizes using ToN_IoT datasets

Batch Size	Accuracy	M-Precision	M-Recall	F1-Score
32	0.9999	0.9999	0.9998	0.9998
64	0.9999	0.9998	0.9995	0.9997
128	0.9999	0.9998	0.9997	0.9997
256	0.9998	0.9998	0.9997	0.9998

Table 15 Performance of proposed ABCNN on different batch sizes using CIC-IDS2017 datasets

Batch Size	Accuracy	M-Precision	M-Recall	F1-Score
32	0.9952	0.9318	0.9359	0.9329
64	0.9938	0.9295	0.9358	0.9216
128	0.9941	0.9295	0.9321	0.9282
256	0.9951	0.9307	0.9301	0.9304

approach achieves an impressive average accuracy of 99.81% on all the datasets. Additionally, other metrics such as precision, recall, and F1-score reach 98.02%, 98.18%, and 98.08% respectively, surpassing the performance of other models and highlighting the superior performance of the proposed model. The proposed model has successfully enhanced the performance of existing systems. There are several classes of attacks available in various IDS datasets. They need to be combined into a single dataset, and the proposed model should be trained by adjusting the hyperparameters. This is because the proposed model has the capability to learn a wider range of attack classes, eliminating the need for multiple models for detecting different cyber attacks. Similar to the limitations of other ML and 1D-DL models, it also requires a feature extraction method to eliminate negative impactful features. Additionally, as the number of layers increases, so does the detection time.

Table 16 Performance comparison with several ML and DL approaches on Edge-IIoTset dataset

Approach	Accuracy	M-Precision	M-Recall	F1-Score	Testing Time (in sec)
LSTM [56]	0.9986	0.9912	0.9672	0.9778	114
CNN [42]	0.9992	0.9935	0.9838	0.9882	20
GRU [57]	0.9991	0.9874	0.9863	0.9865	86
AEC [58]	0.9891	0.9714	0.9735	0.9723	15
LR [45, 47]	0.9933	0.9439	0.8731	0.8829	0.35
NB [47]	0.9961	0.9598	0.9322	0.9402	0.57
SVM [45, 47]	0.9952	0.9827	0.9855	0.9841	150.3
ABCNN	0.9994	0.9921	0.9953	0.9936	27

Table 17 Performance comparison with several ML and DL approaches on IoTID20 dataset

Approach	Accuracy	M-Precision	M-Recall	F1-Score	Testing Time (in sec)
LSTM [56]	0.9911	0.9897	0.9891	0.9893	48
CNN [42]	0.9942	0.9931	0.9938	0.9934	8
GRU [57]	0.9918	0.9884	0.9885	0.9884	38
AEC [58]	0.9971	0.9958	0.9834	0.9881	4
LR [45, 47]	0.8652	0.8107	0.7651	0.7665	0.2
NB [47]	0.7384	0.7735	0.8192	0.7387	0.3
SVM [45, 47]	0.8189	0.8388	0.8189	0.8211	79
ABCNN	0.9981	0.9973	0.9964	0.9969	11

Table 18 Performance comparison with several ML and DL approaches on ToN_IoT dataset

Approach	Accuracy	M-Precision	M-Recall	F1-Score	Testing Time (in sec)
LSTM [56]	0.9995	0.9991	0.9989	0.999	12
CNN [42]	0.9996	0.9996	0.9997	0.9996	2
GRU [57]	0.9991	0.9987	0.9985	0.9986	15
AEC [58]	0.9993	0.9991	0.9995	0.9992	2
LR [45, 47]	0.9943	0.9871	0.9879	0.9875	0.9
NB [47]	0.9952	0.9891	0.9899	0.9896	0.89
SVM [45, 47]	0.9995	0.9992	0.9977	0.9984	15
ABCNN	0.9999	0.9999	0.9998	0.9998	3

Table 19 Performance comparison with several ML and DL approaches on ToN_IoT dataset

Approach	Accuracy	M-Precision	M-Recall	F1-Score	Testing time (in sec)
LSTM [56]	0.9934	0.9251	0.9237	0.9139	113
CNN [42]	0.9902	0.9306	0.9027	0.9111	6
GRU [57]	0.9931	0.9901	0.8109	0.8356	118
AEC [58]	0.9924	0.9231	0.8262	0.8599	6
LR [45, 47]	0.9193	0.6514	0.5743	0.583	0.4
NB [47]	0.7582	0.6761	0.8593	0.6533	0.6
SVM [45, 47]	0.9452	0.8284	0.6206	0.6449	1251
ABCNN	0.9952	0.9318	0.9359	0.9329	8

Author contributions Asadullah Momand: Original Draft, Conceptualization, Methodology, and Investigation. Asadullah Momand played a key role in designing the research framework, formulating the methodology, and conducting experiments. He contributed significantly to data collection and analysis. Sana Ullah Jan: Writing – Review & Editing, Visualization, and Formal Analysis. Sana Ullah Jan took the lead in drafting the initial manuscript, ensuring clarity, coherence, and adherence to academic standards. He actively participated in creating visualizations and conducted formal analyses of the results. Naeem Ramzan: Supervision, Validation, and Writing – Review & Editing. Naeem Ramzan provided oversight throughout the research process, offering valuable guidance and supervision. He critically reviewed and edited the manuscript, ensuring its intellectual rigor, accuracy, and compliance with journal guidelines.

Funding Not Applicable.

Data Availability The following publicly available datasets are used in this study. Edge-IoTset available at: <https://iee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset-iiot-and-iiot-applications>. IoTID20 available at: <https://sites.google.com/view/iiot-network-intrusion-dataset/home>. ToN_IoT available at: <https://research.unsw.edu.au/projects/toniot-datasets>. CIC-IDS2017 available at: <https://www.unb.ca/cic/datasets/ids-2017.html>.

Declarations

Conflict of interest The authors declare no competing interests.

Ethical Approval Not Applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Raza, M. A., Abolhasan, M., Lipman, J., Shariati, N., Ni, W., & Jamalipour, A. (2023). Statistical learning-based adaptive network access for the industrial internet-of-things. *IEEE Internet of Things Journal*.

2. Zhao, R., Huang, Y., Deng, X., Shi, Y., Li, J., Huang, Z., Wang, Y., & Xue, Z. (2023). A novel traffic classifier with attention mechanism for industrial internet of things. *IEEE Transactions on Industrial Informatics*.
3. Niu, S., Shao, H., Yun, S., & Wang, C. (2023). Efficient heterogeneous signcryption scheme based on edge computing for industrial internet of things. *Journal of Systems Architecture*, 136, 102836.
4. Singh, J., Sharma, K., Wazid, M., & Das, A. K. (2023). SINN-RD: Spline interpolation-envisioned neural network-based ransomware detection scheme. *Computers and Electrical Engineering*, 106, 108601.
5. Babar, M., Tariq, M. U., Alshehri, M. D., Ullah, F., & Uddin, M. I. (2022). Smart teledentistry healthcare architecture for medical big data analysis using IoT-enabled environment. *Sustainable Computing: Informatics and Systems*, 35, 100719.
6. Peter, O., Pradhan, A., & Mbohwa, C. (2023). Industrial internet of things (IIoT): Opportunities, challenges, and requirements in manufacturing businesses in emerging economies. *Procedia Computer Science*, 217, 856–865.
7. Babar, M., Jan, M. A., He, X., Tariq, M. U., Mastorakis, S., & Alturki, R. (2023). An optimized IoT-enabled big data analytics architecture for edge-cloud computing. *IEEE Internet of Things Journal*, 10(5), 3995–4005.
8. Calderon-Monge, E., & Ribeiro-Soriano, D. (2023). The role of digitalization in business and management: A systematic literature review. *Review of Managerial Science*, 18(2), 1–43.
9. Pandey, N. K., Kumar, K., Saini, G., & Mishra, A. K. (2023). Security issues and challenges in cloud of things-based applications for industrial automation. *Annals of Operations Research*, 1–20.
10. Mughal, A. A. (2019). Cybersecurity hygiene in the era of internet of things (IoT): Best practices and challenges. *Applied Research in Artificial Intelligence and Cloud Computing*, 2(1), 1–31.
11. Tawalbeh, L., Muheidat, F., Tawalbeh, M., & Quwaider, M. (2020). IoT privacy and security: Challenges and solutions. *Applied Sciences*, 10(12), 4102.
12. Alam, T. (2018). A reliable communication framework and its use in internet of things (IoT). *CSEIT1835111 Received*, 10, 450–456.
13. Chen, M., & Du, W. (2023). The predicting public sentiment evolution on public emergencies under deep learning and internet of things. *The Journal of Supercomputing*, 79(6), 6452–6470.
14. Dal Mas, F., Massaro, M., Rippan, P., & Secundo, G. (2023). The challenges of digital transformation in healthcare: An interdisciplinary literature review, framework, and future research agenda. *Technovation*, 123, 102716.
15. Liang, W., Hu, Y., Zhou, X., Pan, Y., Kevin, I., & Wang, K. (2021). Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT. *IEEE Transactions on Industrial Informatics*, 18(8), 5087–5095.
16. Jeyaselvi, M., Dhanaraj, R. K., Sathya, M., Memon, F. H., Krishnasamy, L., Dev, K., Ziyue, W., & Qureshi, N. M. F. (2023). A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks. *Cluster Computing*, 26(1), 559–574.
17. Xingxin, C., Xin, Z., & Gangming, W. (2022). Research on online fault detection tool of substation equipment based on artificial intelligence. *Journal of King Saud University-Science*, 34(6), 102149.
18. Elsayed, R. A., Hamada, R. A., Abdalla, M. I., & Elsaid, S. A. (2023). Securing IoT and SDN systems using deep-learning based automatic intrusion detection. *Ain Shams Engineering Journal*, 14(10), 102211.
19. Khanday, S. A., Fatima, H., & Rakesh, N. (2023). Implementation of intrusion detection model for DDoS attacks in lightweight IoT networks. *Expert Systems with Applications*, 215, 119330.
20. Jamal, M. H., Khan, M. A., Ullah, S., Alshehri, M. S., Almakdi, S., Rashid, U., Alazeb, A., & Ahmad, J. (2023). Multi-step attack detection in industrial networks using a hybrid deep learning architecture. *Mathematical Biosciences and Engineering*, 20(8), 13824–13848.
21. Rashid, M. M., Khan, S. U., Fariha Eusufzai, Md., Redwan, A., Sabuj, S. R., & Elsharief, M. (2023). A federated learning-based approach for improving intrusion detection in industrial internet of things networks. *Network*, 3(1), 158–179.
22. Misra, S., Pal, S., Ahmed, N., & Mukherjee, A. (2023) SDN-controlled resource-tailored analytics for healthcare IoT system. *IEEE Systems Journal*.
23. Ullah, S., Ahmad, J., Khan, M. A., Alshehri, M. S., Boulila, W., Koubaa, A., Jan, S. U., & Ch, M. M. I. (2023). TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT networks. *Computer Networks*, 237, 110072.
24. Sáez-de Cámara, X., Flores, J. L., Arellano, C., Urbieto, A., & Zurutuza, U. (2023). Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks. *Computers & Security*, 131, 103299.

25. Ullah, S., Boulila, W., Koubaa, A., & Ahmad, J. (2023). MAGRU-IDS: A multi-head attention-based gated recurrent unit for intrusion detection in IIoT networks. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2023.3324657>
26. Gao, Z. J., Pansare, N., & Jermaine, C. (2018). Declarative parameterizations of user-defined functions for large-scale machine learning and optimization. *IEEE Transactions on Knowledge and Data Engineering*, 31(11), 2079–2092.
27. Ullah, S., Khan, M. A., Ahmad, J., Jamal, S. S., e Huma, Z., Hassan, M. T., Pitropakis, N., & Buchanan, W. J. (2022). HDL-IDS: A hybrid deep learning architecture for intrusion detection in the internet of vehicles. *Sensors*, 22(4), 1340.
28. Khan, N. W., Alshehri, M. S., Khan, M. A., Almakdi, S., Moradpoor, N., Alazeb, A., Ullah, S., Naz, N., & Ahmad, J. (2023). A hybrid deep learning-based intrusion detection system for IoT networks. *Mathematical Biosciences and Engineering*, 20(8), 13491–13520.
29. Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20), 4396.
30. Chen, X., Zhang, H., Celimuge, W., Mao, S., Ji, Y., & Bennis, M. (2018). Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet of Things Journal*, 6(3), 4005–4018.
31. Fatani, A., Dahou, A., Abd Elaziz, M., Al-Qaness, M. A., Lu, S., Alfadhli, S. A., & Alresheedi, S. S. (2023). Enhancing intrusion detection systems for IoT and cloud environments using a growth optimizer algorithm and conventional neural networks. *Sensors*, 23(9), 4430.
32. Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 1–19.
33. Al-Turaiki, I., & Altwaijry, N. (2021). A convolutional neural network for improved anomaly-based network intrusion detection. *Big Data*, 9(3), 233–252.
34. Altunay, H. C., & Albayrak, Z. (2023). A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal*, 38, 101322.
35. Wu, J., Dai, H., Wang, Y., Ye, K., & Xu, C. (2023). Heterogeneous domain adaptation for IoT intrusion detection: A geometric graph alignment approach. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2023.3239872>
36. Javadpour, A., Pinto, P., Ja'fari, F., & Zhang, W. (2023). DMAIDPS: A distributed multi-agent intrusion detection and prevention system for cloud IoT environments. *Cluster Computing*, 26(1), 367–384.
37. Thakkar, A., & Lohiya, R. (2023). Attack classification of imbalanced intrusion data for IoT network using ensemble learning-based deep neural network. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2023.3244810>
38. Alghanam, O. A., Almobaideen, W., Saadeh, M., & Adwan, O. (2023). An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. *Expert Systems with Applications*, 213, 118745.
39. Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, 99, 107810.
40. Emec, M., & Özcanhan, M. H. (2022). A hybrid deep learning approach for intrusion detection in IoT networks. *Advances in Electrical and Computer Engineering*, 22(1), 3–12.
41. Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023). Anomaly based network intrusion detection for IoT attacks using deep learning technique. *Computers and Electrical Engineering*, 107, 108626.
42. El-Ghamry, A., Darwish, A., & Hassanien, A. E. (2023). An optimized CNN-based intrusion detection system for reducing risks in smart farming. *Internet of Things*, 22, 100709.
43. Riyaz, B., & Ganapathy, S. (2020). A deep learning approach for effective intrusion detection in wireless networks using CNN. *Soft Computing*, 24(22), 17265–17278.
44. Scarpa, G., Gargiulo, M., Mazza, A., & Gaetano, R. (2018). A CNN-based fusion method for feature extraction from sentinel data. *Remote Sensing*, 10(2), 236.
45. Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, 10, 40281–40306.
46. Kang, H., Ahn, D. H., Lee, G. M., Yoo, J. D., Park, K. H., & Kim, H. K. (2019). *Iot network intrusion dataset*.
47. Ullah, I., & Mahmoud, Q. H. (2020). A scheme for generating a dataset for anomalous activity detection in IoT networks. In *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial*

- Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*, pp. 508–520. Springer.
48. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). Ton_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8, 165130–165150.
 49. Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. *International Journal of Engineering & Technology*, 7(3.24), 479–482.
 50. Stiawan, D., Idris, M. Y. B., Bamhdi, A. M., Budiarto, R., et al. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access*, 8, 132911–132921.
 51. Dahouda, M. K., & Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9, 114381–114391.
 52. Ullah, S., Ahmad, J., Khan, M. A., Alkhamash, E. H., Hadjouni, M., Ghadi, Y. Y., Saeed, F., & Pitropakis, N. (2022). A new intrusion detection system for the internet of things via deep convolutional neural network and feature engineering. *Sensors*, 22(10), 3607.
 53. Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. [arXiv:1212.5701](https://arxiv.org/abs/1212.5701)
 54. Lydia, A., & Francis, S. (2019). AdaGrad-an optimizer for stochastic gradient descent. *International Journal of Computer Science*, 6(5), 566–568.
 55. Kingma, D. P. & Ba, J. (2014). ADAM: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
 56. Alimi, K. O. A., Ouahada, K., Abu-Mahfouz, A. M., Rimer, S., & Alimi, O. A. (2022). Refined LSTM based intrusion detection for denial-of-service attack in internet of things. *Journal of Sensor and Actuator Networks*, 11(3), 32.
 57. Ansari, M. S., Bartoš, V., & Lee, B. (2022). GRU-based deep learning approach for network intrusion alert prediction. *Future Generation Computer Systems*, 128, 235–247.
 58. Altaha, M., Lee, J.-M., Aslam, M., & Hong, S. (2021). An autoencoder-based network intrusion detection system for the SCADA system. *Journal of Communication*, 16(6), 210–216.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Asadullah Momand was born in Nangarhar Afghanistan. Growing up in a culturally rich environment, he developed an early interest in Computing. His formative years were marked by a voracious appetite for learning, which led him to pursue higher education with a focus on Machine Learning. Asadullah Momand is Currently Doing a PhD from University of the West of Scotland. He has done his Master's in Computing from Edinburgh Napier University in 2020. He has authored more than 3 papers in international journals and peer-reviewed international conference proceedings. His research area is closely related to the Artificial Intelligence or Machine Learning-based cyber security and privacy in the Internet-of-Things.



Dr. Sana Ullah Jan is an experienced teacher and researcher with a history of serving in prestigious institutes including the University of the West of Scotland, the University of Ulsan (South Korea) and the University of Lahore (Pakistan). He is currently enrolled as Lecturer/Assistant Professor in Edinburgh Napier University, UK since September 2021. He was previously enrolled as Post-doctoral Research Fellow at the Center of Affective and Human Computing for Smart Environment at the school of computing, engineering and physical sciences, University of the West of Scotland since September 2020 to August 2021. He has (co)authored more than 30 papers in international journals and peer-reviewed international conference proceedings. His research area is closely related to the Artificial Intelligence or Machine Learning-based cyber security and privacy in the Internet-of-Things, Cyber Physical Systems and eHealth. He has taught various courses both at Undergraduate (UG) and Postgraduate (PG) levels during his career. He is serving as editorial board member and invited reviewer for several leading high-impact journals and conferences. He has been

endorsed as Global Talent by the Royal Academy of Engineering of the UK. He is a senior member of IEEE and a fellow of Higher Education Academy (HEA) of the UK.



Professor Naeem Ramzan (S'04, M'08, SM'13) received the M.Sc. degree in telecommunication from University of Brest, France, in 2004 and the Ph.D. degree in electronics engineering from Queen Mary University of London, London, U.K., in 2008. Currently, he is a Full Professor of Computer Engineering in University of West of Scotland. Before that he was a senior research fellow and lecturer at Queen Mary University of London from 2008 to 2012. He is a Director of Artificial Intelligence Virtual Communication and Network (AVCN) institute and Chair of Affective and Human Computing for Smart Environment (AHCSE) Research Centre. He is, a Fellow of Royal Society of Edinburgh, senior member of the IEEE Fellow, Senior Fellow of Higher Education Academy (HEA), co-chair of MPEG HEVC verification (AHG5) group and a voting member of the British Standard Institution (BSI). In addition, he holds key roles in the Video Quality Expert Group (VQEG) such as Co-chair of the Ultra High Definition (UltraHD) group; Co-chair of the Visually Lossless Quality Analysis (VLQA) group; and Co-chair of the Psycho-Physiological Quality

Assessment (PsyPhyQA). He has been a lead researcher in various nationally or EU sponsored multimillion funded international research projects. His research interests are cross-disciplinary & industry focused and include: video processing, analysis and communication, video quality evaluation, Brain-inspired multi-modal cognitive technology, Big Data analytics, Affective computing, IoT/smart environments, natural multi-modal human computer interaction, eHealth/connected Health. He has a global collaborative research network spanning both academia and key industrial players. He has been the Lead supervisor/supervisor for about 20 postdoctoral research fellows and PhD research students, and six PhD students supervised by him, have successfully completed in UK. He has published more than 150 articles in peer reviewed journals, conferences, book chapters including standardised contributions. His paper was awarded best paper award 2016 of IEEE Transaction of Circuit and System for Video Technology and three conference papers were selected for best student paper award in 2015/2016. He has been awarded STARS (Staff Appreciation and Recognition Scheme) award for 2014 and 2016 for "Outstanding Research and Knowledge Exchange" (University of the West of Scotland) and Awarded Contribution Reward Scheme 2011 and 2009 for outstanding research and teaching activities (Queen Mary University of London). Currently he is a Co-Editor-in-Chief of VQEG eLetter and served as guest editor in a number of journals. He is a Founding Associate Editor of Springer Journal "Quality and User Experience" and Associate Editor of number of Journals. He has chaired/co-chaired/organised more than 25 workshops, special sessions, and tracks in International conferences.