Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Securing IoT Networks: An Intrusion Detection Framework Using Convolutional Neural Networks

AHMAD Bello[1], MAGAJI Yusuf Mada[2]
[1]Department of Computer Science, [2]Department of Computer Engineering
[1, 2]Abdu Gusau Polytechnic Talata Mafara, Zamfara State
belomada38@gmail.com[1], Yusufmagajimada12@gmail.com[2]

Abstract

The proliferation of the Internet of Things (IoT) has significantly transformed human lives, embedding interconnected devices across various sectors such as industry, healthcare, agriculture, and transportation. However, this growth has also attracted cybercriminals, leading to a surge in cyber-attacks targeting IoT devices. Intrusion Detection Systems (IDS) have emerged as a critical defense mechanism against such attacks. In this paper, we propose a Convolutional Neural Network (CNN)-based framework for IoT intrusion detection. The framework employs CNNs for automatic feature selection from raw network traffic data, enhancing classification performance and reducing dimensionality. We utilize the IoTID20 benchmark dataset for training and testing the proposed framework. Experimental results demonstrate the effectiveness of our approach, achieving high accuracy of 99.93%, precision of 1.00%, recall of 99.92%, and F1-score of 99.98% for binary classification. For multi-class category classification, the framework achieves an accuracy of 99.51%, precision of 99.98%, recall of 99.87%, and F1-score of 99.93%. For multi-class sub-category classification, the accuracy is 92.46%, precision is 97.25%, recall is 90.54%, and F1-score is 92.83%. Our proposed framework outperforms existing methods, making it a promising solution for IoT network security.

Keywords: *Internet of Things, Cyber-attacks, Intrusion Detection Systems, Convolutional Neural Network, IoT Network security*

1. Introduction

The rapid proliferation of the **Internet of Things (IoT)** has revolutionized human lives, connecting a multitude of devices and enabling seamless communication, accumulating and processing of data (Farooq et al., 2022). From smart homes to industrial automation, healthcare, and supply chain management, IoT applications have become ubiquitous. This type of network is increasingly becoming an essential part of our everyday life and is providing a variety of applications in industries, medical, agriculture, businesses, and intelligent transportation. The use of IoT devices has drastically increased dramatically, from 15.41 billion in 2015 to more than 35.8 billion in 2021, as devices and businesses increasingly rely on online technology (Alam, 2018). The IoT is anticipated to reach 75.44 billion devices by 2025, as indicate in Figure 1, which will generate 79 zettabytes (ZB) of data (Al-Bahri et al., 2018). IoT has been identified as a critical component of digitization for a transforming society (Nguyen et al., 2022). However, this exponential growth has also attracted the attention of cybercriminals, leading to an alarming surge in cyber-attacks targeting IoT devices and their communication channels. The vulnerable devices could be later used by adversaries to join an IoT botnet and participate in widespread attacks. For example, the IoT botnet that was launched in October 2016, called Mirai (Antonakakis et al., 2017), was able to compromise those vulnerable CCTV cameras that were using default usernames and passwords to launch a DDoS attack on DNS servers. This attack resulted in stopping the Internet accessibility in some parts of the USA. In April 2020, an IoT botnet, named Mozi, was discovered and was found capable of launching various DDoS

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

attacks (Fadilpasic, 2020; Vijayan, 2020). Therefore, building security systems is very necessary, and many industrial and academic organizations have developed different systems and solutions.

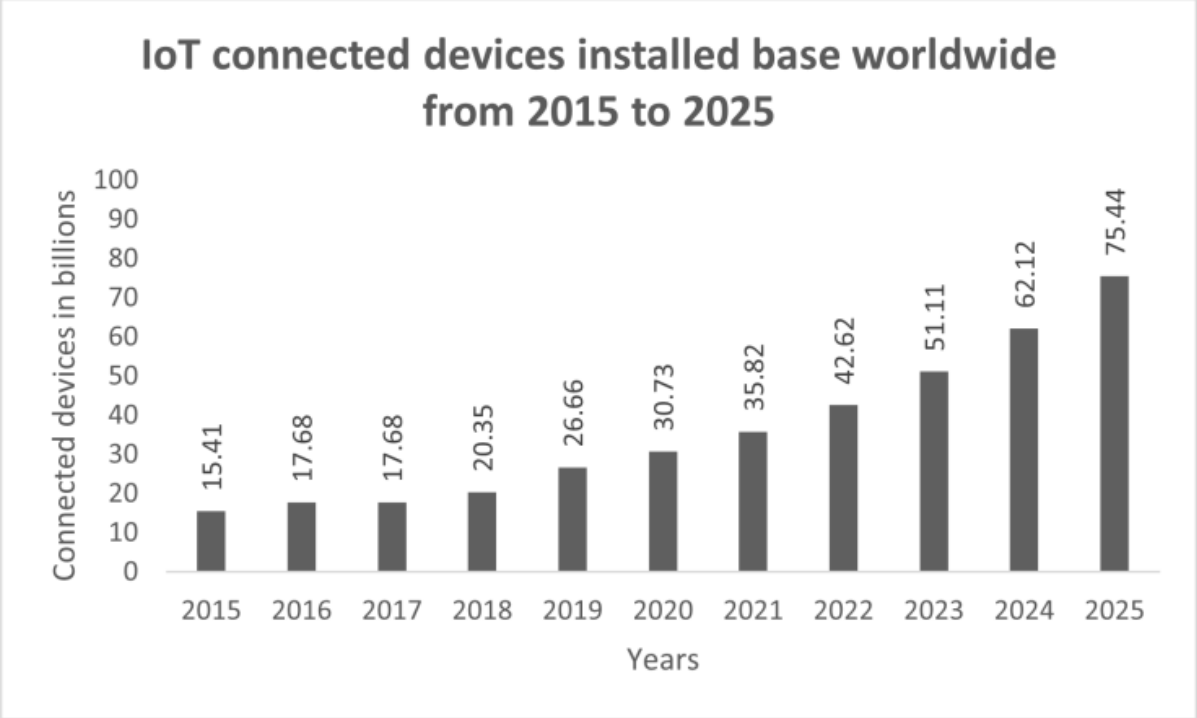


Figure 1. IoT devices from 2015 to 2025 (Al-Bahri et al., 2018)

IoT networks are still vulnerable to network assaults in spite of the availability of conventional security measures, so designing a second line of defense is necessary to identify potential attackers. Intrusion Detection Systems (IDS) has gained a significant consideration among the best security mechanisms for safeguarding the IoT cyber infrastructures against various cyber-attacks in the last decades. **It** plays a pivotal role in safeguarding IoT networks. Detecting and mitigating malicious activities in real-time is crucial for ensuring the reliability, security, and profitability of IoT-enabled services. In short, IDS are becoming a common solution for improving security in Internet of Things (IoT) networks. The three most well-known types of security techniques to find anomalies and intrusions in IoT devices are the signature or misuse-based IDS, anomaly-based IDS, and hybrid approaches (Behniafar, Nowroozi, & S., 2020). IDSs are divided into two categories: i) intrusion detection systems that are based on hosts (HIDSs) and ii) intrusion detection systems that are based on networks (NIDSs). The later captures and analyzes packet flows in the network. In other words, they are scanning sniffed packets while the former is placed on host machine. Conventional (IDS) often fall short due to the dynamic nature of IoT environments, diverse communication protocols, and the sheer volume of data generated by connected devices.

Various Machine Learning Algorithms have been used in the past to develop methods for IDS, k-means clustering (Zhao & Zhang, 2016), decision trees (DT) (Modi et al., 2012), k-nearest neighbor (kNN) (Ghosh et al., 2015), support vector machines (SVM) (Wei et al., 2020), and other traditional machine learning (ML) approaches, In recent years, **Deep Learning (DL)** has emerged as a powerful tool for enhancing intrusion detection capabilities. DL models, particularly neural networks, excel at learning complex patterns from large-scale data. Leveraging DL techniques can significantly improve the accuracy and efficiency of IoT intrusion detection systems. Convolutional neural networks (CNNs) (Asgharzadeh et al., 2023) is one of the promising approach to detecting IoT-based attacks. However, the work in (Ullah et

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

al., 2022) proposed a DCNN technique for malicious activity identification in IoT networks. The works uses Extra Tress Classify (ETC) for the feature selection, which is an ensemble learning method that combines multiple decision trees for feature selection in classification and feature selection tasks, it struggle to model complex, non-linear interactions, and do not provide such feature maps, making it harder to explain why specific features were selected. It also cannot provide clear insights into individual feature importance due to random threshold-based splits, potentially causing a loss of feature importance information.

One significant technique for monitoring IoT threats is feature selection. From the massive amounts of data generated by IoT devices, important attribute must be selected in order to identify illicit behavior. The aim of this method is to determine the most pertinent metrics and features that help detect IoT network attack. The Accurate and efficient of the IoT-based threats detection model relies heavily on feature selection, which enhances the data quality for training models and speeds up the detection process. Using feature selection approaches, discriminability and relevancy are the criteria used to choose features for IoT threat detection. Choosing attributes that are closely linked to the behaviors and characteristics of IoT-based attacks is known as relevance. This allows for the meaningful collection of data that helps differentiate between benign and malevolent activity in IoT networks. To effectively distinguish between various attacks or between attacks and typical network behavior, the features chosen must be discriminable (Alabsi et al., 2023). CNNs can be deployed to the dataset to select important attributes that are useful in the identification of IoT-based threats. CNNs can be able to identify and rank features that are most important for identifying IoT-based attacks by utilizing their ability to learn complex patterns and representations from dataset. The efficacy and accuracy of IoT threat detection model adding to the minimum computational cost expansiveness are improved by this feature selection procedure.

2. Related Works

In this section, we provide an overview of some previous proposed approaches of IDS in IoT. An approach in (Rihan, Anbar, & Alabsi, 2023) *for detecting* attacks on *IoT networks using* ensemble feature *selection* and *deep learning* models *using* Recursive Feature Elimination (RFE) is proposed, and the impact of the selected feature set on the per*for*mance of *Deep Learning* models (*CNN*, RNN, GRU, and LSTM) is evaluated *using* the *IoT*-Botnet 2020 dataset. Although, they achieve promising result, but the model suffers in computational overhead and requires more parameters to optimize. The work in (Dahou et al., 2022) a new feature selection mechanism is proposed based on a recently developed MH method, called Reptile Search Algorithm (RSA), which is inspired by the hunting behaviors of the crocodiles, which boosts the IDS system performance by selecting only the most important features from the extracted features using the CNN model.

(Khan et al., 2022) The study presents an autoencoder-based detection framework for identifying cyber threats in Industrial Internet of Things (IIoT) networks. It uses convolutional and recurrent networks and a two-step sliding window approach to learn latent representations of data features. Malicious data points are transformed into fixed-length series and continuous-time-reliant subseries, capturing temporal aspects of malicious events. The framework effectively extracts features, including malicious patterns' contexts, and outperforms current methods, making it suitable for real-world IIoT networks.

(Khan & Ullah, 2022) A new malware detection framework, Deep Squeezed-Boosted and Ensemble Learning (DSBEL), is developed for the Internet of Things (IoT) environment. The framework uses a novel Squeezed-Boosted Boundary-Region Split-Transform-Merge (SB-BR-STM) convolutional neural network (CNN) and ensemble learning to capture both homogeneous and heterogeneous global malicious patterns. Transfer learning and multi-path-based squeezing and boosting are used to achieve

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

diverse feature maps and learn minute pattern variations. The framework's effectiveness is demonstrated using the IOT_Malware dataset, achieving high accuracy, F1-Score, MCC, recall, and precision. The framework is robust for timely detection of malicious activities.

(Bhosale & Sridhar Patnaik, 2022) The study introduces a lightweight Deep Learning model, LDC-Net, for COVID-19 case identification using chest X-ray images. The model uses a Deep Convolutional Neural Network to train, overcoming common challenges like overfitting and generalization errors. Experimental results on Raspberry Pi show promising results, with an overall precision of 96.86%, recall of 96.78%, F1-score of 96.77%, and accuracy of 99.28%. This framework can be deployed on IoT and IoMT devices to identify COVID-19 cases, other lung diseases, and healthy labels.

Another study proposes a deep learning-based approach for heart disease detection using IoT-enabled electrocardiogram (ECG) devices. The methodology involves several steps: i. Collection of IoT-based ECG signals from both healthy individuals and individuals with heart disease. ii. Preprocessing of ECG signals using finite impulse response. iii. Feature extraction including P-wave, ST-segment, R-peak locations, heart rate variability, ST-segment, PQ-segment, T-wave, QRS complex duration, continuous wavelet transform, and improved mutual information. iv. Feature selection using a new hybrid optimization model called the alpha spider customized dwarf mongoose optimizer, which combines the standard tunicate swarm algorithm and slime mould algorithm. v. Heart disease detection using a new three-layer framework comprising convolutional neural networks, bidirectional long-short term memory, and recurrent neural networks. The performance of the proposed methodology is evaluated using performance metrics such as accuracy, sensitivity, specificity, precision, recall, false positive rate (FPR), and false negative rate (FNR). The methodology is implemented in MATLAB platform (Mishra & Tiwari, 2023).

(Alamro et al., 2023) The study proposes a Blockchain Assisted IoT Healthcare System using Ant Lion Optimizer with Hybrid Deep Learning (BHS-ALOHDL) technique to improve security and privacy of medical data in IoT healthcare systems. The technique combines blockchain technology with intrusion detection systems to create a robust security framework. The technique uses Ant Lion Optimizer for feature subset selection and a Hybrid Deep Learning model for intrusion detection. The flower pollination algorithm is used for optimal hyperparameter tuning, improving detection rate. Experimental results show promising performance compared to other models.

The study presents a framework to improve the performance of applications in the Social Internet of Things (SIoT) system, focusing on big data management, analysis, and reporting. The framework consists of two main components: Transformer CNN (TransCNN) for feature extraction and Chaos Game Optimization (CGO) algorithm for feature selection. The method was tested on various datasets and compared with other existing methods. The results showed that the proposed method outperformed other models in the SIoT environment, with an average accuracy of 88.30%, sensitivity of 87.20%, specificity of 92.94%, 44.375 features selected, and a fitness value of 0.1082. The mean rank obtained using the Friedman test indicates the best overall performance compared to competitive algorithms (Dahou, Chelloug, Alduailij, & Elaziz, 2023).

The (Odeh & Abu Taleb, 2023) study explores IoT intrusion detection using deep learning models like CNNs, LSTM, and GRUs. It develops an ensemble deep learning framework that incorporates a voting policy for hierarchical pattern computation. The ensemble deep learning classifiers outperform traditional techniques, with CNN-LSTM and CNN-GRU models achieving 99.7% and 99.6% accuracy respectively. The developed method has an average accuracy of 88.30%, sensitivity of 87.20%, specificity of 92.94%, selects 44.375 features, and a fitness value of 0.1082, indicating the best overall performance compared to competitive algorithms.

The paper presents a deep-convolutional-neural-network (DCNN)-based Intrusion Detection System (IDS) for IoT networks, aiming to improve performance and reduce computational power. The

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

model, consisting of two convolutional layers and three dense layers, was tested using the IoTID20 dataset and evaluated using metrics like accuracy, precision, recall, and F1-score. Various optimization techniques were applied, with Adam, AdaMax, and Nadam showing the best performance. The model was compared with advanced deep learning and traditional machine learning techniques, achieving high accuracy and robustness. The method achieved a fitness value of 0.1082, making it the best overall compared to competitive algorithms (Ullah et al., 2022).

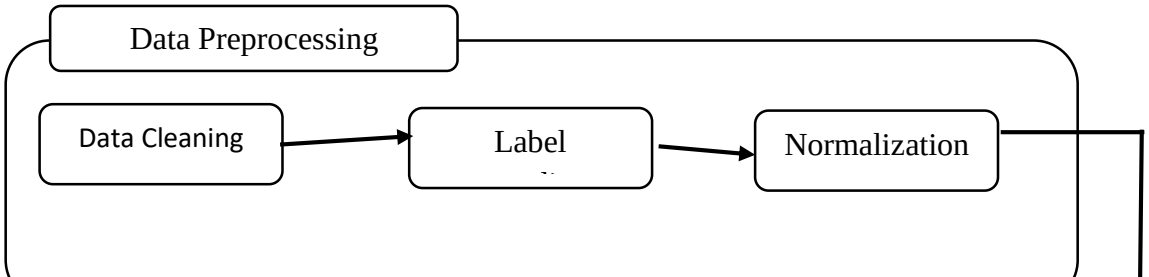
3. Methodology

In this section, we present our proposed framework, convolutional neural network architecture for feature selection. Our framework aims to leverage the power of CNN to automatically select relevant features from IoT network traffic data. The most informative features selected by CNN can then be used to accurately classify network traffic as normal or an attack using a fully connected dense layer.

Data preprocessing

One crucial step in machine learning and deep learning workflows is data preprocessing, which entails transforming raw data into a format that is appropriate for model training. Improving the quality of the data and making it more suitable for the learning algorithm is the goal of the process, which in turn boosts the model's performance (Alasadi & Bhaya, 2017). To achieve this, several steps were followed in the proposed approach. To begin with, dataset cleaning, where the dataset must be verified for empty and undefined instances before training a model, The utilized IoTID20 benchmark dataset has some missing values. To clean the dataset, we removed all missing value instances. In addition, label encoding is a well-known encoding approach for dealing with categorical values. It assigns a unique numeric value to each categorical value. The dataset has some categorical features. Each categorical feature has several categories. A label encoder approach was used to convert the categorical features into numeric. Furthermore, normalization is a method commonly used in the preprocessing of data that converts the numeric column values in a dataset to a common scale while maintaining variations in value ranges. The data are normalized between 0 and 1 via the min-max method, as represented by Equation 1. Last but not least, for data splitting, we use a stratified method to split data into train and test sets for the model, addressing the issue of unequal data splits in unbalanced datasets. The method splits the entire dataset into homogenous sets, with 75% of the data split into train sets and 25% for test sets for each class, ensuring an accurate evaluation of the model's performance.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad 1$$



Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Figure 2. Architecture of the proposed framework.

Designing framework

This is the main stage of the proposed framework; it aims to select features automatically from the row data using a 1D CNN. This enhances classification performance and reduces dimensionality. By leveraging the hierarchical and adaptive properties of CNNs for complex datasets with high dimensionality, this approach enables precise and efficient attribute selection. Additionally, in order to avoid manual feature engineering, we used CNNs for feature selection, which allows us to automatically learn the most informative features from the input data. Employing CNN for feature selection, we first train the CNN model using the provided IoTID20 dataset. Then, we retrieve the last convolutional layer from the trained model, equation 2 and 3 present the convolution equation for convolution and pooling layers respectively. The outcome of the last pooling layer is converted into a single-dimensional array by the flatten layer, which is the input of the first dense layer. The output of the first dense layer is given as input in the second dense layer. The second dense layer produces output, which is input in the last dense layer, and the output of the last dense layer is considered the classification result. The activation function (rectify linear unit) is used in dense layers. The last dense layer produces output results in which softmax and sigmoid activation functions are used for multi-class and binary classifications, respectively. Equation 4, 5 and 6 represent the ReLU, Softmax and Sigmoid activation functions respectively.

$$y_m^{i,y} = f\left(b_m^{i,j} + \sum_{s=1}^{s} w_{s,m}^{i,j} * l_{m+s-1}^{i-1,j}\right) \qquad 2$$

where, the output of the m^{th} neuron in layer i is represented as $y_m^{i,y}$, the function $f()$ indicates the activation function, the offset of the neuron for layer i is represented as $b_m^{i,j}$ and the output of the neuron for layer i - 1 is represented as $l_{m+s-1}^{i-1,j}$ and the convolution kernel for i^{th} layer is represented as $w_{s,m}^{i,j}$. Followed by each

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

convolution layer, a pooling layer is included in the architecture which reduces the dimensions of the features obtained from the convolution layer. This pooling operation reduces the network complexity by reducing the dimension and avoids data overfitting.

$$\varphi_m^{i,j} = f\left(\delta_m^{i,j} pool\left(l_m^{i-1,j}\right) + b_m^{i,j}\right) \quad 3$$

where, the m^{th} neuron output for layer i is represented as $\varphi_m^{i,j}$, the function $f()$ indicates the activation function, the offset of the neuron for layer i is represented as $b_m^{i,j}$, the sampling weight coefficient is represented as $\delta_m^{i,j}$, the pooling function is represented as $pool()$ and the output of the i - 1 is represented as $\delta_m^{i,j}$. The process of convolution and pooling is performed three times as the architecture has three sets of convolution and pooling layers.

$$f(x_k) = max(0, x_k) \quad 4$$

$$softmax(x)_i = \frac{e^{x_i}}{\sum_{j-1}^{k} e^{x_j}} \quad 5$$

$$\delta(x) = \frac{1}{1+e^{-x}} \quad 6$$

Table 1. Architecture of the CNN used.

Layer (type)	Output Shape	Number of parameters
conv1d_3 (Conv1D)	(None, 81, 64)	256
max_pooling1d_3 (MaxPooling1D)	(None, 40, 64)	0
conv1d_4 (Conv1D)	(None, 38, 32)	6176
max_pooling1d_4 (MaxPooling1D)	(None, 19, 32)	0
conv1d_5 (Conv1D)	(None, 17, 16)	1552

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

max_pooling1d_5 (MaxPooling1D)	(None, 8, 16)	0
flatten_1 (Flatten)	(None, 128)	0
dense_3 (Dense)	(None, 128)	16512
dense_4 (Dense)	(None, 64)	8256
dense_5 (Dense)	(None, 1)	65

Utilizing Convolutional Neural Networks (CNNs) as a feature selection technique offers a distinct advantage by surpassing the limitations of traditional filter and wrapper selection methods. CNNs have an inherent ability to learn hierarchical representations from input data, allowing them to autonomously extract relevant features without relying on external input or predefined criteria. As a result, CNNs have the potential to uncover correlations and patterns that may go unnoticed by conventional feature selection techniques.

IoTID20 dataset

The IoTID20 dataset is a benchmark dataset designed for evaluating intrusion detection systems (IDSs) in Internet of Things (IoT) networks presented in (Ullah & Mahmoud, 2020). It contains network traffic data captured from a real-world IoT environment, specifically a smart home setting. The dataset includes both normal and attack traffic, making it suitable for training and testing IDSs. The dataset captures network traffic from a smart home IoT environment, providing a realistic representation of IoT network behavior and covering a variety of IoT attack scenarios, allowing for comprehensive testing of IDSs. Network traffic is captured using packet capturing tools, providing detailed information about packet payloads, source and destination IP addresses, protocols used, etc. The dataset is designed to facilitate the evaluation of IDSs in IoT networks, allowing researchers to assess the effectiveness of different intrusion detection techniques. Table 2 show the distribution of IoTID20 dataset.

Table 2. The row distribution of IoTID20 dataset

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

No. of normal category records	40,073
No. of attack category records	585,710
Total no. of records	625,783
No. of features	86

Table 3. IoTID20 Dataset Feature set and data types

Sl.	Features	Data types	Sl.	Features	Data types	Sl.	Features	Data types
1	Flow_ID	object	30	Fwd_IAT_Max	float64	59	Pkt_Size_Avg	float64
2	Src_IP	object	31	Fwd_IAT_Min	float64	60	Fwd_Seg_Size_Avg	float64
3	Src_Port	int64	32	Bwd_IAT_Tot	float64	61	Bwd_Se_Size_Avg	float64
4	Dst_IP	object	33	Bwd_IAT_Mean	float64	62	Fwd_Byts/b_Avg	int64
5	Dst_Port	int64	34	Bwd_IAT_Std	float64	63	Fwd_Pkts/b_Avg	int64
6	Protocol	int64	35	Bwd_IAT_Max	float64	64	Fwd_Blk_Rate_Avg	int64
7	Timestamp	'0'	36	Bwd_IAT_Min	float64	65	Bwd_Byts/b_Avg	int64
8	Flow_Duration	int64	37	Fwd_PSH_Flags	int64	66	Bwd_Pkts/b_Avg	int64
9	Tot_Fwd_Pkts	int64	38	Bwd_PSH_Flags	int64	67	Bwd_Blk_Rate_Avg	int64
10	Tot_Bwd_Pkts	int64	39	Fwd_URG_Flags	int64	68	Subflow_Fwd_Pkts	int64
11	TotLen_Fwd_Pkts	float64	40	Bwd_URG_Flags	int64	69	Subflow_Fwd_Byts	int64
12	TotLen_Bwd_Pkts	float64	41	Fwd_Header_Len	int64	70	Subflow_Bwd_Pkts	int64
13	Fwd_Pkt_Len_Max	float64	42	Bwd_Header_Len	int64	71	Subflow_Bwd_Byts	int64
14	Fwd_Pkt_Len_Min	float64	43	Fwd_Pkts/s	float64	72	Init_Fwd_Win_Byts	int64
15	Fwd_Pkt_Len_Mean	float64	44	Bwd_Pkts/s	float64	73	Init_Bwd_Win_Byts	int64
16	Fwd_Pkt_Len_Std	float64	45	Pkt_Len_Min	float64	74	Fwd_Act_Data_Pkts	int64
17	Bwd_Pkt_Len_Max	float64	46	Pkt_Len_Max	float64	75	Fwd_Seg_Size_Min	int64
18	Bwd_Pkt_Len_Min	float64	47	Pkt_Len_Mean	float64	76	Active_Mean	float64
19	Bwd_Pkt_Len_Mean	float64	48	Pkt_Len_Std	float64	77	Active_Std	float64
20	Bwd_Pkt_Len_Std	float64	49	Pkt_Len_Var	float64	78	Active_Max	float64
21	Flow_Byts/s	float64	50	FIN_Flag_Cnt	int64	79	Active_Min	float64
22	Flow_Pkts/s	float64	51	SYN_Flag_Cnt	int64	80	Idle_Mean	float64
23	Flow_IAT_Mean	float64	52	RST_Flag_Cnt	int64	81	Idle_Std	float64
24	Flow_IAT_Std	float64	53	PSH_Flag_Cnt	int64	82	Idle_Max	float64
25	Flow_IAT_Max	float64	54	ACK_Flag_Cnt	int64	83	Idle_Min	float64
26	Flow_IAT_Min	float64	55	URG_Flag_Cnt	int64	84	Label	int64
27	Fwd_IAT_Tot	float64	56	CWE_Flag_Count	int64	85	Cat	Object
28	Fwd_IAT_Mean	float64	57	ECE_Flag_Cnt	int64	86	Sub_Cat	Object
29	Bwd_IAT_Mean	float64	58	Down/Up_Ratio	float64	-	-	-

4. Results and discussion

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

In our experiments, Adam (Kingma & Ba, 2014) optimizer is used to update the CNN model weights and bias using a default learning rate of 0.001. The CNN model was trained for 100 epochs using a 64 batch size. The exceptional performance scores tabulated in Table 4 confirm that a CNN can be leveraged to identify significant features that contribute to detecting IoT attacks. Additionally, the performance of the framework was tested for binary, multi-class categories, and multi-class subcategories classifications. In binary classification, the binary cross-entropy function was used to calculate the loss and for multi-class classification, categorical sparse cross-entropy was used.

The results from the experiment are presented in Tables 4 where four evaluation metrics, which are: Accuracy, Precision, Recall, and F1-score were measured so as to evaluate the potential of the proposed approach while detecting an anomaly in IoT networks. The details on how these metrics are computed can be viewed in the study conducted in (Maniriho et al., 2020). If the precision is high, it results in low false positive alarms, which greatly reduces false alerts triggered by the detection model. Besides, for the anomalous traffic, precision is always measured. Recall indicates the proportion of actual positives instances that were correctly identified. The last metric is the F1-score, which measure of a test's accuracy, and it considers both the precision and the recall of the test to compute the score.

Table 4**.** Performance comparison with DCNN in (Ullah et al., 2022).

Classification		**Accuracy**	**Precision**	**Recall**	**F1-score**
Binary	DCNN	99.84	99.67	99.02	99.34
	Our CNN	99.93	1.00	99.92	99.98
Multi class category	DCNN	98.12	97.13	97.83	97.46
	Our CNN	99.51	99.98	99.87	99.93
Multi class sub-category	DCNN	77.55	78.76	73.43	76.00
	Our CNN	92.46	97.25	90.54	92.83

As it was previously mentioned, the proposed method was trained and tested using the IoTID20 dataset. Percentage split were employed to split the dataset into training and testing sets. The split is 75% for training and 25% for testing the Deep Learning approach. From Table 4, it can be observed that for binary classification, the accuracy, precision, recall and f1-score of the proposed method has the scores of 99.93%, 1.00%, 99.92% and 99.98% respectively. On the other hand, scores for multi-class category is 99.51 for accuracy, 99.87% for precision, 99.87 for recall and f1-score of 99.93% were recorded. Finally, by observing the outcome from the evaluation metrics, this model has also managed to classify multi-class sub-category as shown in the table 4 above.

The overall classification accuracy for each classification as presented in Table 4 where the accuracy for binary classification (99.93%), Multi-class category (99.51%), multi-class sub-category (99.46%) were

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

obtained using percentage split. To evaluate how well the proposed model performs compared to the previous ones, Figure. 3, 4, and 5 indicates our method's performance compared to the one in (Ullah et al., 2022), while classifying binary, multi-class category and multi-class sub-category where the proposed method outdoes the previous one in terms of accuracy 99.93% over 99.84% for binary, 99.51% over 98.12% for multi-class category and 92.46% over 77.55% for multi-class sub-category. Note that the dataset used in (Ullah et al., 2022) is the same with the one used in this work. Considering the overall results, the proposed framework can be deployed in IoT networks to detect various IoT network attacks.

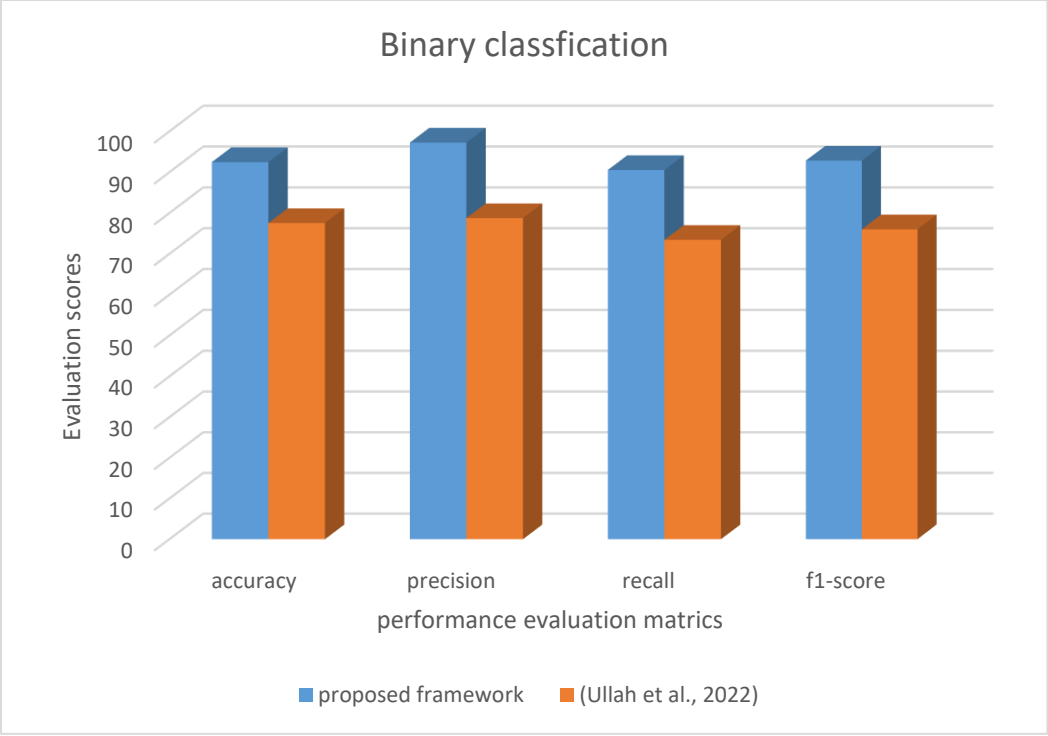


Figure 3. Binary classification comparison.

Kasu Journal of Computer Science Vol. 1 No. 2 [June, 2024], pp. 218-233
https://doi.org//10.47514/kjcs/2024.1.2.006
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

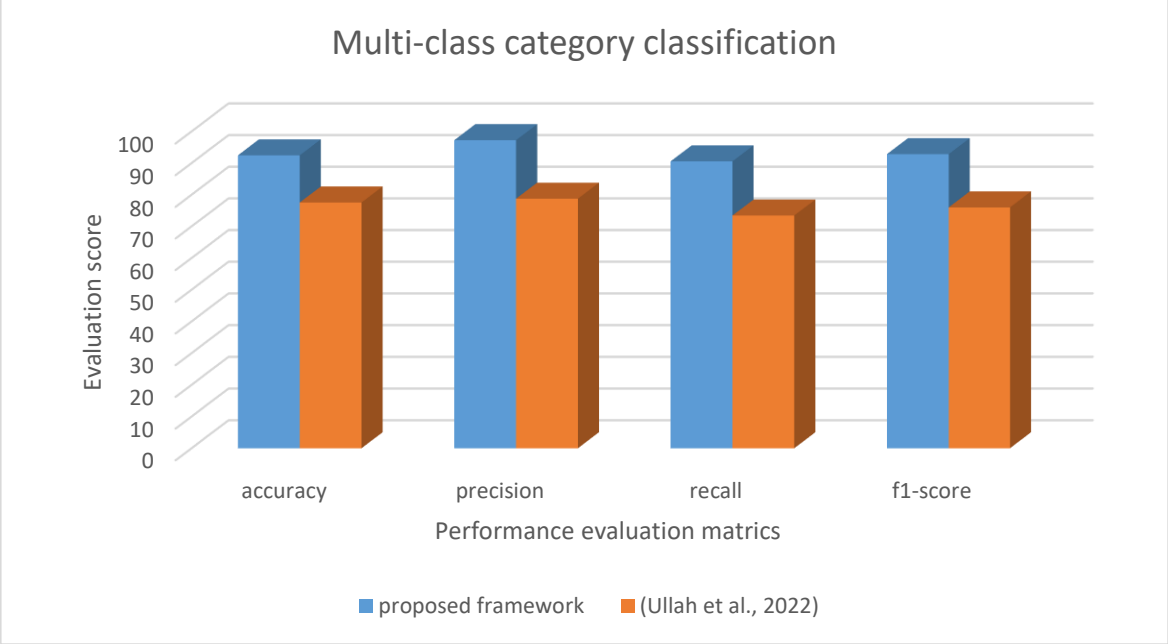


Figure 4. Multi-class category classification comparison.

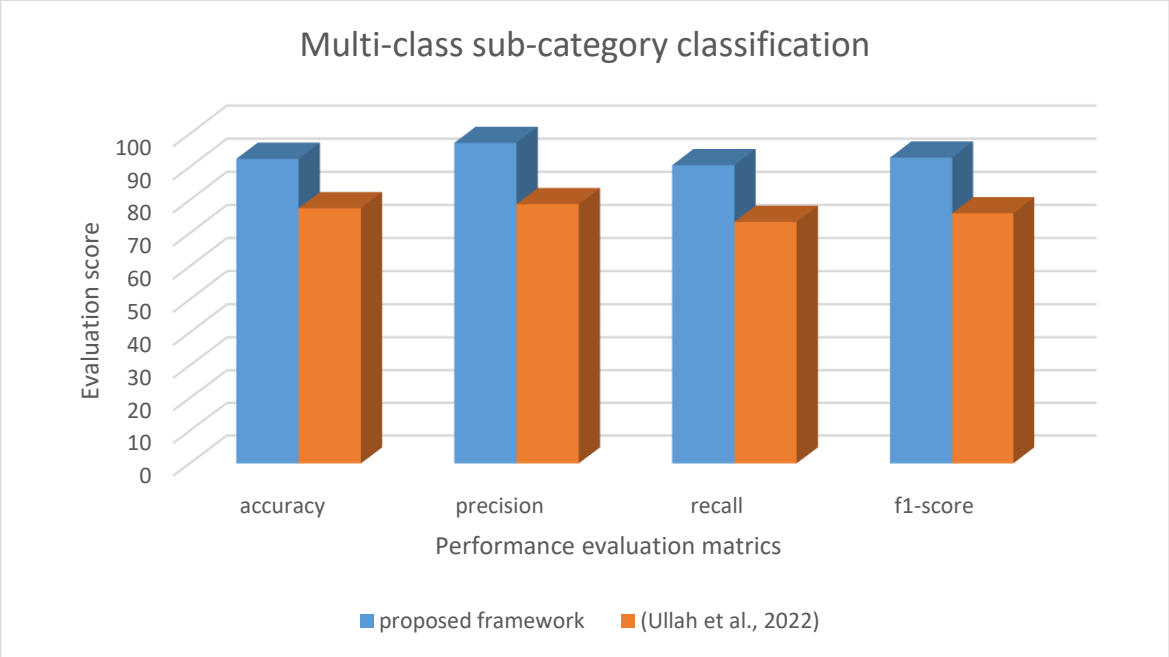


Figure 5. Multi-class sub-category classification comparison

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

5. Conclusion

In this study, we proposed a CNN-based framework for IoT intrusion detection, aiming to improve the accuracy and efficiency of intrusion detection systems in IoT networks. Leveraging the hierarchical and adaptive properties of CNNs, our framework automatically selects relevant features from raw network traffic data, enhancing classification performance and reducing dimensionality. Experimental results using the IoTID20 benchmark dataset demonstrate the effectiveness of our approach, achieving high accuracy, precision, recall, and F1-score for binary, multi-class category, and multi-class sub-category classifications. Our framework outperforms existing methods, indicating its potential for deployment in real-world IoT networks to detect various IoT network attacks. Future research directions could explore the integration of additional deep learning techniques and optimization strategies to further improve the framework's performance and scalability.

Reference

Alam, T. (2018). A Reliable Communication Framework and Its Use in Internet of Things (IoT). *International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 3,* 450–456.

Al-Bahri, M., Yankovsky, A., Borodin, A., & Kirichek, R. (2018). Testbed for identifying IoT devices based on digital object architecture. In Internet of Things, Smart Spaces, and Next Generation Networks and Systems (pp. 129–137). Springer.

Alabsi, B.A., Anbar, M., & Rihan, S.D.A. (2023). CNN-CNN: Dual Convolutional Neural Network Approach for Feature Selection and Attack Detection on Internet of Things Networks. Sensors, 23, 6507. https://doi.org/10.3390/s23146507

Alamro, H., Marzouk, R., Alruwais, N., Negm, N., Aljameel, S.S., Khalid, M., Hamza, M.A., & Alsaid, M.I. (2023). Modeling of Blockchain Assisted Intrusion Detection on IoT Healthcare System Using Ant Lion Optimizer With Hybrid Deep Learning. IEEE Access, 11, 82199-82207.

Antonakakis, M., April, T., Bailey, M., et al. (2017). Understanding the Mirai botnet. In 26th USENIX Security Symposium (USENIX Security17) (pp. 1093–1110). Vancouver, BC, Canada.

Behniafar, M., Nowroozi, A., & H. R. S. (2020). A survey of anomaly detection approaches in Internet of Things. The ICS INT'L Journal of Information Security, 10(2), 79–92.

Bhosale, Y.H., & Sridhar Patnaik, K. (2022). IoT Deployable Lightweight Deep Learning Application For COVID-19 Detection With Lung Diseases Using RaspberryPi. 2022 *International Conference on IoT and Blockchain Technology (ICIBT*), 1-6.

Dahou, A., Abd Elaziz, M.A., Chelloug, S.A., Awadallah, M.A., Al-Betar, M.A., Al-qaness, M.A., & Forestiero, A. (2022). Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm. Computational Intelligence and Neuroscience, 2022.

Dahou, A., Chelloug, S.A., Alduailij, M.A., & Elaziz, M.E. (2023). Improved Feature Selection Based on Chaos Game Optimization for Social Internet of Things with a Novel Deep Learning Model. Mathematics.

Fadilpasic, S. (2020, April). Researchers discover IoT botnet capable of launching various DDoS attacks. ITProPortal. https://www.itproportal.com/news/researchers-discover-iot-botnet-capable-of-launching-

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

various-ddos-attacks/.

Farooq, M.S., Sohail, O.O., Abid, A., & Rasheed, S. (2022). A Survey on the Role of IoT in Agriculture for the Implementation of Smart Livestock Environment. *IEEE Access*, 10, 9483–9505.

Ghosh, P., Mandal, A.K., & Kumar, R. (2015). An efficient cloud network intrusion detection system. In Information Systems Design and Intelligent Applications (pp. 91–99). Springer.

Kingma, D.P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980.

Khan, D.H., & Ullah, W. (2022). A New Deep Boosted CNN and Ensemble Learning based IoT Malware Detection. Comput. Secur., 133, 103385.

Khan, I.A., Moustafa, N., Pi, D., Sallam, K.M., Zomaya, A., & Li, B. (2022). A New Explainable Deep Learning Framework for Cyber Threat Discovery in Industrial IoT Networks. *IEEE Internet of Things Journal, 9,* 11604-11613.

Maniriho, P., Mahoro, L. J., Niyigaba, E., Bizimana, Z., & Ahmad, T. (2020). Detecting Intrusions in Computer Network Traffic with Machine Learning. *International Journal of Intelligent Engineering and Systems, 13(3),* 433-445.

Mishra, J., & Tiwari, M. (2023). IoT-enabled ECG-based heart disease prediction using three-layer deep learning and meta-heuristic approach. Signal, Image and Video Processing, 18, 361-367.

Modi, C., Patel, D., Borisanya, B., Patel, A., & Rajarajan, M. (2012). A novel framework for intrusion detection in cloud. *In Proceedings of the fifth International Conference on Security of Information and Networks (pp. 67–74).* Jaipur, India.

Nguyen, X.H., Nguyen, X.D., Huynh, H.H., & Le, K.H. (2022). Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways. Sensors, 22, 432.

Odeh, A.J., & Abu Taleb, A. (2023). Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection. Applied Sciences.

Rihan , S.D., Anbar , M., & Alabsi, B.A. (2023). Approach for Detecting Attacks on IoT Networks Based on Ensemble Feature Selection and Deep Learning Models. Sensors (Basel, Switzerland), 23.

Ullah, I., & Mahmoud, H. Q. (2020). A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In 33rd Canadian Conference on Artificial Intelligence.

Ullah, S., Ahmad, J., Khan, M.A., Alkhammash, 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, 3607. https://doi.org/10.3390/s22103607.

Wei, J., Long, C., Li, J., & Zhao, J. (2020). An intrusion detection algorithm based on bag representation with ensemble support vector machine in cloud computing. Concurrency and Computation: *Practice and Experience*, 32, e5922.

Zhao, X., & Zhang, W. (2016). An anomaly intrusion detection method based on improved k-means of cloud computing. *In Proceedings of the 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)* (pp. 284–288). Harbin, China.