


Article

A Comprehensive Security Framework for Asymmetrical IoT Network Environments to Monitor and Classify Cyberattack via Machine Learning

Ali Alqahtani ^{1,*} , Abdulaziz A. Alsulami ² , Nayef Alqahtani ³ , Badraddin Alturki ⁴ 
and Bandar M. Alghamdi ⁴ 

¹ Department of Networks and Communications Engineering, College of Computer Science and Information Systems, Najran University, Najran 61441, Saudi Arabia

² Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; aaalsulami10@kau.edu.sa

³ Department of Electrical Engineering, College of Engineering, King Faisal University, Al-Ahsa 31982, Saudi Arabia; nmalqahtani@kfu.edu.sa

⁴ Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; baalturki@kau.edu.sa (B.A.); bmmalghamdi@kau.edu.sa (B.M.A.)

* Correspondence: asalqahtany@nu.edu.sa

Abstract: The Internet of Things (IoT) is an important component of the smart environment, which produces a large volume of data that is considered challenging to handle. In addition, the IoT architecture is vulnerable to many cyberattacks that can target operational devices. Therefore, there is a need for monitoring IoT traffic to analyze, detect malicious activity, and classify cyberattack types. This research proposes a security framework to monitor asymmetrical network traffic in an IoT environment. The framework offers a network intrusion detection system (NIDS) to detect and classify cyberattacks, implemented using a machine learning (ML) model residing in the middleware layer of the IoT architecture. A dimensionality reduction technique known as principal component analysis (PCA) is utilized to facilitate data transmission, which is intended to be sent from the middleware layer to the cloud layer with reduced complexity and fewer unnecessary inputs without compromising the information content. Therefore, the reduced IoT traffic data are sent to the cloud and the PCA data are retransformed to approximate the original data for visualizing the IoT traffic. The NIDS is responsible for reporting the attack type to the cloud in the event of an attack. Our findings indicate that the proposed framework has promising results in classifying the attack type, which achieved a classification accuracy of 98%. In addition, the dimension of the IoT traffic data is reduced by around 50% and it has a similarity of around 90% compared to the original data.

Keywords: NIDS; IoT; cyberattack; CIC-IoT 2023 Dataset; machine learning; deep learning; PCA



Citation: Alqahtani, A.; Alsulami, A.A.; Alqahtani, N.; Alturki, B.; Alghamdi, B.M. A Comprehensive Security Framework for Asymmetrical IoT Network Environments to Monitor and Classify Cyberattack via Machine Learning. *Symmetry* **2024**, *16*, 1121. <https://doi.org/10.3390/sym16091121>

Academic Editor: Jie Yang

Received: 28 July 2024

Revised: 17 August 2024

Accepted: 26 August 2024

Published: 29 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things (IoT) has started reaching maturity levels with the evolution of sensors, data analytics, and high capabilities of devices that are connected to the Internet. There are many application domains such as smart grid, microgrid, and smart industries, and other domains that depend on various devices for their realization such as smartphones, routers, and sensors. These applications utilize devices that sense and preprocess data including data fusion and filtering tasks locally at the core of the infrastructure of communication, while most of the high-capability data analysis tasks are executed in the cloud. The IoT provides improved productivity and adaptability, which helps in developing frameworks that are highly interconnected and lead to new services being enabled [1]. The IoT paradigm has entered the market of the industry with tailored solutions in the past decades, thereby the concept and the idea of industrial IoT and industry 4.0 have been defined [2]. The projections of the number of IoT devices will be around 32 billion Internet-connected

devices, which can be considered as two times the number of Internet-connected devices in 2024. IoT devices are utilized in both the consumer and corporate sectors [3]. There are many projections regarding the number of Internet-connected devices, the numbers might vary but the important aspect here is that the number of Internet-connected devices is increasing tremendously. This means that many devices will be connected to the Internet, which will bring many security issues. The networks in IoT are open, and the topology frequently changes in real time as nodes join and leave the network. This means that they are susceptible to cybersecurity threats due to a shortage of centralized management for networks. Every IoT device has characteristics such as small memory size, constrained storage and power, and limited bandwidth [4]. In IoT infrastructure, these limitations in IoT devices can impact on the effectiveness of cybersecurity protocols including performance and growth.

Therefore, monitoring IoT network traffic and detecting and classifying cyberattacks targeting IoT architecture can be considered challenging because of the overhead that needs high power of computation. Recently, hackers or attackers started using cutting-edge strategies to attack and obtain information, which makes cyberattacks more complex for IDS to detect. The IoT can be connected with anyone or anything, which means that any user (authorized or not authorized) can interact with the IoT, including sensors, actuators, protocols, and cloud IoT devices inside the IoT network. In an IoT network, it is difficult to make decisions on the content, whether it is malicious or normal because it is challenging to detect the characteristics of the content. Figure 1 illustrates that the attacks can happen mostly within the networks or between the IoT devices and the cloud in the network. The figure also shows the architecture of IoT, which mostly consists of four layers named IoT devices, IoT gateways, middleware, and cloud platform. IoT devices collect data from the operating environment via sensors, and the collected data are sent to the middleware through IoT gateways. The middleware layer utilizes data analytics and management. The cloud has high performance, which can store and handle large amounts of data [5,6].

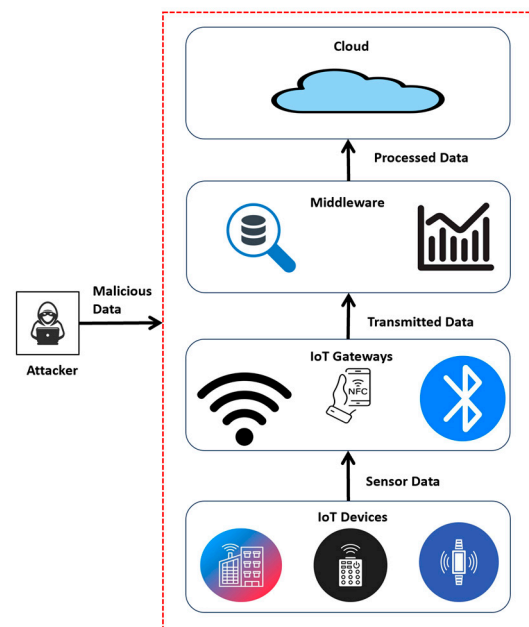


Figure 1. IoT general architecture.

As a result, there is a need for innovative and effective methods for real-time network monitoring and network intrusion detection systems. In the last decade, researchers started using machine learning and deep learning methods to detect and prevent abnormality and intrusions [7]. The IoT paradigm allows the integration of sensors and objects, which enables the exchange of data. The networks of IoT have been efficient in gathering, an-

analyzing, reporting, and forecasting data to be used in the future [8]. One of the earliest definitions of IoT is that “The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service” [9]. In addition, the IoT and Internet help to perform more tasks with fewer resources and less time. The definition can be updated by adding intelligence to the IoT because IoT in the current era is not just a connection between humans and things.

IoT needs include many aspects such as data analytics, security monitoring for the network, and intrusion detection systems for the network. This indicates that there are evolvments in IoT technology. According to the authors in [10], there is an increase in the number of attacks that impact IoT systems. It can be considered that IoT-based systems are more vulnerable to cyber threats and various attacks targeting these systems. As the number of Internet-connected devices increases, cyberattacks will also increase, which indicates that maintaining IoT security is crucial [11]. There will be intrusions in the IoT system, and the intrusion detection system (IDS) is a solution that can be hardware or software that is designed to detect intrusions of activities while maintaining security. The authors in [12] state that in cloud and IoT environments, it is essential to utilize intrusion detection and prevention systems (IDPSs) for securing such environments. The objective of IDS is to detect device usage that is unauthorized and monitor network traffic for malicious activity that is traditionally captured with a firewall. There are two groups of IDS including anomaly-based detection systems (AIDS) and signature-based detection systems (SIDS) [13]. AIDS can detect new attacks effectively while SIDS is effective with the known threats [14]. There are many network intrusion detection frameworks proposed particularly deep learning frameworks considered as common [15]. There are many challenges in IoT security, and one of the crucial ones is its variety of vulnerabilities.

As the IoT networks expand, the cybersecurity challenges will increase and that will threaten the security of the connected things and devices [16]. The cybersecurity challenges include vulnerabilities in IoT devices and things firmware, insecure communication protocols, and the challenges of developing effective measures of security within devices that are resource-constrained [1]. IoT devices are being integrated into many infrastructures including smart industries, smart grids, smart homes, and smart healthcare, this will impact the cybersecurity attacks to grow. These attacks focus on IoT environments that might cause unauthorized access, possibly physical damage, and data breaches [17], which shows that there is a need for an effective cybersecurity framework.

There are several methods to exploit vulnerabilities and assess the susceptibility of devices in IoT [18]. Most of the attacks are executed using connectivity to the Internet and network, but sometimes there might be physical cyberattacks, which are not out of the scope of this study. It is known that IoT devices are constrained, thereby utilizing old-fashioned security techniques may not be an effective choice for securing the vulnerable environment of IoT [19]. In an IoT environment, the devices produce tremendously high volumes of data, which makes the process of monitoring and detecting intrusions through these streams challenging. Utilizing machine learning and deep learning methods can be effective in addressing this challenge. Therefore, this research proposes a security framework that is used with asymmetric IoT architecture. The framework enables monitoring of IoT traffic, which is assumed to be sent from the middleware layer to the cloud layer. The PCA technique is used to reduce the dimensionality of the data in the middleware layer before it is transmitted to the cloud. In the cloud layer, the PCA components are retransformed to approximate the original data, which will be monitored and visualized. In addition, the proposed framework uses a network intrusion detection system (NIDS), implemented based on ML to detect and classify the cyberattack type, and then the result is reported to the cloud.

The contribution of the paper can be summarized as follows:

- A PCA technique is used to reduce the dimensionality of data without compromising the information content. Therefore, the reduced IoT data are sent to the cloud for monitoring and visualizing the IoT traffic.

- The inverse PCA is computed to retransform the PCA components to approximate the original data for visualizing the IoT traffic.
- The framework offers an NIDS to detect and classify cyberattacks, which is responsible for sending notifications with the attack type to the cloud in the event of an attack.
- Comparing the performance of NIDS implemented based on various ML models, which are Long Short-Term Memory, Gated Recurrent Unit (GRU), 1D Convolutional Neural Networks (1D-CNN), Feedforward Neural Network (FFNN), Decision Tree (DT), and Random Forest (RF).

The rest of the paper is organized as follows. In Section 2, the related works are reviewed, followed by Section 3, which discusses methodology, dataset, and experimental setup. Then, the results and findings are presented and discussed. Finally, the conclusion is drawn, and the future possible works are mentioned.

2. Literature Review

In this section, the related works in the field of IoT cybersecurity will be presented by reviewing existing works and exploring the works that used ML and DL methods. Also, the works focused on intrusion detection systems and explored the challenges that can exist without the use of IDS. Therefore, this section will discuss the IoT security challenges, intrusion detection systems in the IoT, and related works.

The IoT includes a significant number of resources, which is considered as an asymmetric environment such as smart grid, smart industry, and smart city. These environments have asymmetric resource distribution on various sides of devices including the end user side, fog side, and cloud side [20].

The authors in [21] proposed a convolutional neural network (CNN) to improve the security and efficiency of networks in IoT by developing AIDS. Their model has the ability to detect intrusions and anomalies within the network traffic by utilizing the datasets called BoT-IoT [22] and network intrusion detection (NID) [23], they have achieved an accuracy of 92.85% and 99.51%, respectively.

The authors in [24] utilized CNN for feature extraction in an automated way and proposed a model called IoT feature extraction convolutional neural network (IoTfECNN). They designed a version of the capuchin search algorithm (CSA) in an optimized way to select features using binary multi-objective techniques while enhancing efficiency. They evaluated their method using datasets called NSL-KDD and TON-IoT. Their model achieved an accuracy of 99.85% and 99.99% and feature characteristics detected were 27% and 44%. Their results were promising when compared with other existing methods in the literature. The researchers in [25] introduced an IDS framework based on several approaches of recurrent neural networks (RNNs) including simple RNN, GRU RNNs, and LSTM while using XGBoost for feature selection for the datasets called UNSW NB15 and NSL KDD. Their findings show that the best performance is achieved by using the NSL-KDD dataset and XGBoost-LSTM with a test accuracy (TAC) of 88.13% and validation accuracy (VAC) of 99.49%. However, their model achieved a test accuracy of 87.07% with the UNSW-NB15 dataset and the XGBoost-Simple-RNN, and it was the most efficient with that dataset. When it comes to multiclass classification, the XGBoost-LSTM achieved a test accuracy of 86.93% with the dataset of the NSL-KDD. In addition, the XGBoost-GRU has achieved a TAC of 78.40% with the dataset of UNSW-NB15. Based on their results the XGBoost Simple RNN achieved training time shorter than others, which can be considered as a good choice with IoT devices as minimal resources will be used. The researchers in [26] proposed a lightweight Random Neural Network for intrusion detection in the field of IoT. Their method can be considered a good selection for networks in IoT because the framework is distributed and flexible. The performance of their framework is evaluated on single classes in the dataset, but in minority classes, the enhancement is not addressed. Their results show that attack detection has achieved an accuracy of 99.14% in binary and an accuracy of 99.05% in multiclass. They tested their model using a dataset called TON-IoT. Based on the above-mentioned papers, there is a need to improve the rate of intrusion detection in

IoT systems. Also, it is important to detect anomalies using IoT devices, which will help in attack mitigation and security measures enhancement. The researchers did not give full attention to attack type and anomaly type in the field of IoT.

The authors in [27] proposed deep learning-based IDS by using focal loss to evaluate three different datasets from IoT domains. Their results showed that DL models using focal loss have better performance in precision, F1-score, accuracy, and MCC score by 24%, 39%, 39%, and 60%, respectively, when compared to training the models on the datasets used previously by utilizing cross-entropy loss. They have compared their work with other methods in the literature and their results were promising. The researchers in [28] introduced an ensemble learning method aimed to work with imbalance issues and solve them within intrusion detection datasets that utilize deep neural network-based intrusion detection systems. They used a classifier called bagging with a deep neural network to be a base estimator. Their model is evaluated by utilizing four different datasets in intrusion detection including UNSW-NB-15, NSL-KDD, BoT-IoT, and CIC-IDS-2017. Their analysis of the result of their approach is presented utilizing several metrics of evaluation including precision, accuracy, recall, false positive rate, and F1 -score. Furthermore, the results of their method are tested statistically by utilizing the Wilcoxon signed-rank test and the result was promising as they achieved an accuracy of 98.99% with the BoT-IoT dataset. The authors in [29] proposed a model that is based on several deep learning models including LSTM, CNN, and a hybrid of both LSTM and CNN for intrusion detection in the networks of the Industrial Internet of Things (IIoT). They achieved high performance with the hybrid model when applied to two datasets for intrusion detection called X-IIoTID and UNSW-NB15, the accuracy of the model was 92.9% for multiclass and 93.1% for binary class when it was used with the UNSW-NB15 dataset. However, when the model was applied to the X-IIoTID dataset, the accuracy was 99.80% for multiclass and 99.84% for binary class. Researchers in [10] presented an intrusion detection framework for machine learning (IDFML) for the Internet of Things networks. They used supervised learning in their techniques and their model achieved an accuracy of 98.68% while attack detection was performed using a classifier called Random Forest.

The authors in [30] utilized a Fully Connected Feed Forward Neural Network (FCFFN) architecture to detect intrusion in network traffic that is focused on IoT devices. The system utilizes an architecture that is four layers deep and fully connected. They propose a deep learning-based IDS for IoT devices. Their model detects attacks such as DDoS, blackhole, sinkhole, and wormhole on IoT networks with an average accuracy of 93.74%. Their IDS score in average detection is 93.21% rate, which can be considered a good indication for enhancing IoT network security. The researchers in [31] proposed a classifier that is flow-based traffic that utilizes the flow transformer method for anonymity and attacks in IIoT network traffic. Their model uses a multi-head attention method to explore the significance of the flow sequence and utilizes a layer that has feature extraction that works to monitor and capture the sequence's characteristics of the flow. They achieved an accuracy of 98.5% by utilizing the dataset called CIC-IoT-2022. The authors in [32] introduced a deep ensemble-based intrusion detection system that utilizes a multi-classifier system that is lambda architecture, which helps in detecting attacks through classifiers like CNN, ANN, and LSTM to have the maximum layer batch. The model is trained by batch layer, while the decisions are made in the speed layer of lambda architecture to have enhanced real-time evaluation. In binary classification, the LSTM has better accuracy than others. The hybrid ensemble has an accuracy of 99.93% and the results show that the ensemble presents good accuracy of detection. The researchers in [33] introduced anomaly detection that is based on deep learning for the IoT that has the ability to learn and detect robust features that might not be affected by environments that are unstable. The captured features are then utilized by a classifier to help improve the accuracy of the malicious detection process. To have the robustness of features against environments that are heterogeneous in the IoT. Their results show that their framework is effective when it comes to improving the accuracy of malicious detection as they compared their work with related works. The authors in [34]

proposed a model that is based on Mayfly optimization (MFO) with a regularized extreme learning machine (RELM). The model called MFO-RELM is used to detect and classify cybersecurity threats in an IoT environment. Their model can effectively identify cyber threats in the heterogeneous IoT environment. The model preprocesses IoT data to have more understandable information or format. The model is applied to a dataset called N-BaIoT to test and help in identifying the cyber threats in IoT. They achieved an average accuracy of 99.8%.

According to [35], researchers introduced an ensemble model based on deep learning that utilizes the power of LSTM and autoencoder architecture to detect abnormal events for detecting cyber threats in IIoT. The LSTM is used to generate a model based on time series data that has past and present information, which will help when learning the normal pattern of the data. Also, the autoencoder is used to identify the features that are important for data dimension reduction. Furthermore, their model can extract new data that are balanced from a dataset that is imbalanced. The newly extracted balanced data are integrated into the model. They compared the results with classifiers in machine learning such as Random Forest, multi-layer perceptron, Decision Tree, and support vector machine, the model achieved high accuracy compared to them. They achieved an accuracy of 99.3% and 99.7% when applying two datasets called GP and SWaT, respectively. Research by [36] proposed an intrusion detection system that uses a filter-based deep neural network model. They aimed to solve the issue of imbalanced data. To overcome this issue, they increased the packets in attack categories in the minority class within the networks of IoT through generative adversarial networks. Their model achieved an accuracy of 84% in multi-class when applied to a dataset called UNSW-NB15, whereas generative adversarial networks achieved an accuracy of 91% in an accuracy class dataset that is balanced. A study in [37] presented an IDS for the security of fog computing that is automated and utilizes AI for cyberattack mitigation. The proposed model uses multi-layered recurrent neural networks (RNNs) that are developed to stand near the data source for fog computing security. The model has achieved an accuracy of 98.27 when detecting DoS attacks when applied to the dataset called NSL-KDD. Authors in [38] introduced a framework that is based on deep learning for cyberattack detection. For evaluation of the performance purposes of LSTM and FFNN models, they used two datasets including NSL-KDD and BoT-IoT. Their results were promising as they achieved an accuracy of cyberattacks detection rate of 99.95% with LSTM and 99.97% with FFNN. Table 1 shows the summary of the literature review.

Table 1. Summary of literature review.

Study	Techniques	Dataset	Key Findings
[21]	CNN	BoT-IoT and network intrusion detection (NID)	Accuracy of 92.85% and 99.51%
[24]	CNN	NSL-KDD and TON-IoT	99.85% and 99.99%
[25]	Simple RNN, GRU RNNs and LSTM	UNSW NB15 and NSL KDD	NSL-KDD dataset and XGBoost-LSTM with TAC of 88.13% and VAC of 99.49%. TAC of 87.07% with UNSW-NB15 dataset and the XGBoost-Simple-RNN XGBoost-LSTM achieved a TAC of 86.93% with NSL-KDD XGBoost-GRU has achieved a TAC of 78.40% with UNSW-NB15
[26]	Lightweight Random Neural Network	TON-IoT	Accuracy of 99.14% in binary and an accuracy of 99.05% in multiclass

Table 1. Cont.

Study	Techniques	Dataset	Key Findings
		BoT-IoT	
[27]	Feed Forward Neural Networks (FNNs) and Convolutional Neural Networks (CNNs)	WUSTL-EHMS-2020 WUSTL-IIoT-2021	FNN focal: accuracy of 91.55% CNN focal: accuracy of 86.77% FNN focal: accuracy of 98.95% CNN focal: accuracy of 98.21% FNN focal: accuracy of 93.26% CNN focal: accuracy of 93.08%
[28]	Deep Neural Network	NSL-KDD UNSW-NB15 CIC-IDS-2017 BoT-IoT	accuracy of 98.9% accuracy of 96.7% accuracy of 98.74% accuracy of 98.99%
[29]	LSTM and CNN	UNSW-NB15	Accuracy of 92.9% for multiclass and 93.1% for binary class
[10]	Random Forest	IoTID20	Accuracy of 98.68%
[30]	FCFFN	Generated IoT dataset	Accuracy of 93.74%
[31]	Flow Transformer	CIC-IoT-2022	Accuracy of 98.5%
[32]	Ensemble ANN CNN LSTM		Accuracy in batch mode Ensemble: 99.6% ANN: 96.9% CNN: 97.0% LSTM: 98.2
[33]	DNN	DS2OS Traffic	Accuracy of 94.9%
[34]	MFO-RELM	N-BaIoT	Accuracy of 99.8%
[35]	Ensemble LSTM-Auto Encoder (AE)	GP and SWaT	Accuracy of 99.3% and 99.7%
[36]	DNN GAN	UNSW-NB15	DNN: Accuracy of 84% GAN: Accuracy of 91%
[37]	RNN	NSL-KDD	Accuracy of 92.18%
[38]	FFNN LSTM	NSL-KDD BoT-IoT	FFNN: accuracy of 98.67% LSTM: accuracy of 96.44% FFNN: accuracy of 99.97% LSTM: accuracy of 99.95%

In summary, we have reviewed several papers in the literature, some of these papers have a limited scope, which might be a limitation when creating comprehensive intrusion detection systems. The researchers in the literature did not pay full attention to the dimension of data. Also, they did not consider the traffic between IoT devices and cloud computing. Most of the papers have not considered monitoring the network traffic in IoT, but they focused on intrusion detection systems.

3. Methodology

This research proposes an efficient IoT security framework for monitoring network traffic and classifying cyberattacks. It monitors network traffic and offers a network intrusion detection system (NIDS) to detect and classify cyberattacks. Network traffic is transmitted to the cloud after applying the PCA transformation technique to reduce the dimensions of the data, which helps to send data with a smaller size. In the cloud, the PCA components are transformed back to approximate the original data; therefore, system

administrators can monitor that data. In the meantime, the NIDS, which is implemented based on ML, performs detection and classification to find malicious activity and report it to the cloud. The results of NIDS are transmitted to the cloud to show the presence of a cyberattack and its type.

3.1. The Proposed Framework Architecture

The proposed framework of this research is illustrated in Figure 2. The core functionality of the framework is to monitor the IoT traffic and detect and classify cyberattacks. As depicted, IoT devices usually involve sensors that collect different data types, such as temperature, humidity, camera monitoring, etc. In our case, IoT devices are applied to capture real-time data from the IoT network traffic within IoT environments. Subsequently, data preprocessing is performed in order to randomly select sufficient and clean records of the dataset. The sampled dataset is then transmitted in two directions, the first is used for monitoring IoT traffic, and the second direction is used for detecting and classifying cyberattack types.

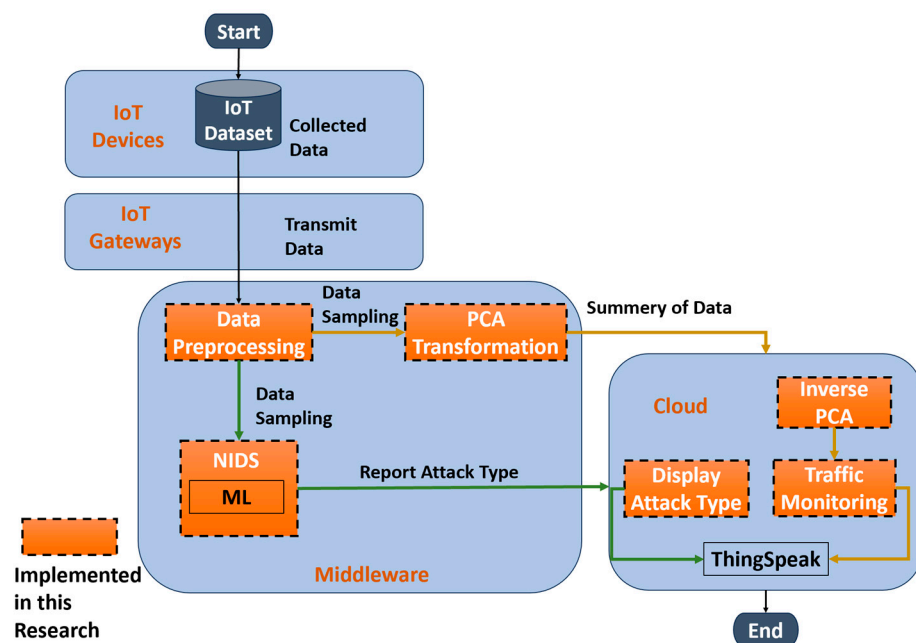


Figure 2. The proposed IoT security framework.

The PCA transformation subsystem is responsible for reducing the dimensions of data, leading to efficient data collection, transmission, and processing. The PCA parameters are then transmitted and once they arrive in the cloud, the inverse PCA subsystem is used to approximate the original data. The inverse PCA subsystem uses a restoration technique that occurs in the cloud and involves an algorithm that inverts the PCA transformation and restores the original data characteristics. Finally, the cloud and IoT data are monitored and visualized using the ThingSpeak platform.

ThingSpeak is an IoT platform that allows streamed data to be visualized and analyzed in the cloud. Therefore, it enables real-time visualizations of data that are produced by connected devices. The data can be sent to ThingSpeak using REST API or Message Queuing Telemetry Transport (MQTT) [39]. Our research utilizes ThingSpeak to aggregate and visualize the IoT traffic through the REST API protocol.

The NIDS subsystem is implemented using an ML model to detect and classify attack types. The sampled data are transmitted to the NIDS as shown in Figure 2 and after completing the detection and classifying procedure, the attack type is sent to ThingSpeak to be viewed by the end user. By integrating IoT cloud-based monitoring and ML classifier,

the proposed system will provide a comprehensive security framework for IoT network traffic monitoring and cyberattack classification.

3.2. IoT Dataset

The Canadian Institute for Cybersecurity published the CIC-IoT 2023 Dataset, which contains real-time IoT traffic [40,41]. The dataset comprises 169 files of IoT network traffic data. The total records for all files are approximately 46,686,579. Each record includes information about 33 classes of different attacks and one benign class. The number of features for each class is 46. Figure 3 shows the original class distribution for each attack. We randomly collected a maximum of 40,000 records of each dataset class and merged them into one file, as shown in Figure 4. It is common in cybersecurity that dataset suffers from imbalance. We have mitigated the imbalance by collecting a maximum of 40,000 records of each dataset and ensured a more equal representation of each attack class. This is a very important preprocessing step to enhance the ML model’s capability to recognize the patterns and effectively classify a wide range of attacks. From Figure 4, we can notice certain attack classes, like DDoS-Slow Loris, Dictionary Brute Force, and DDoS-HTTP Flood have relatively high instance counts, ranging from around 13,000 to 29,000. However, a minimum number of other attack classes, like Backdoor Malware, Browser Hijacking, Command Injection, Recon-PingSweep, SQL Injection, Uploading Attack, and XSS, have lower instance counts, in the low thousands.

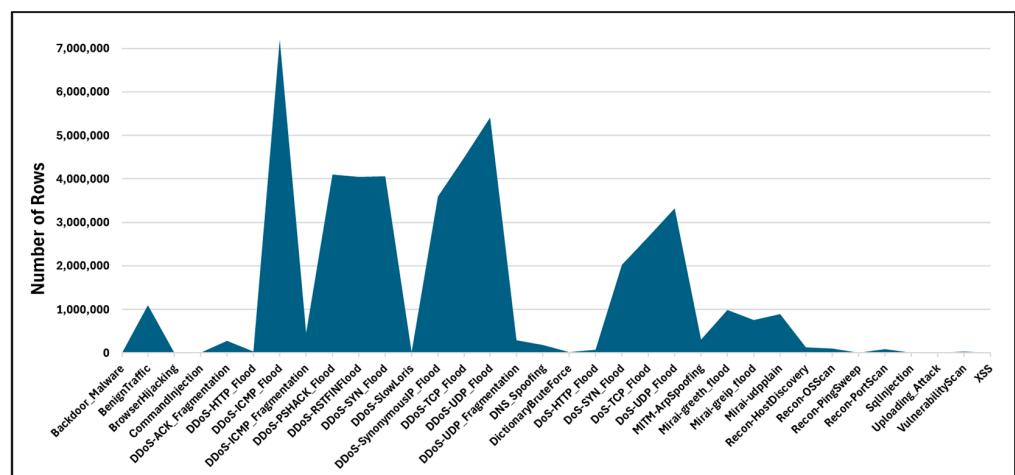


Figure 3. Original dataset distribution.

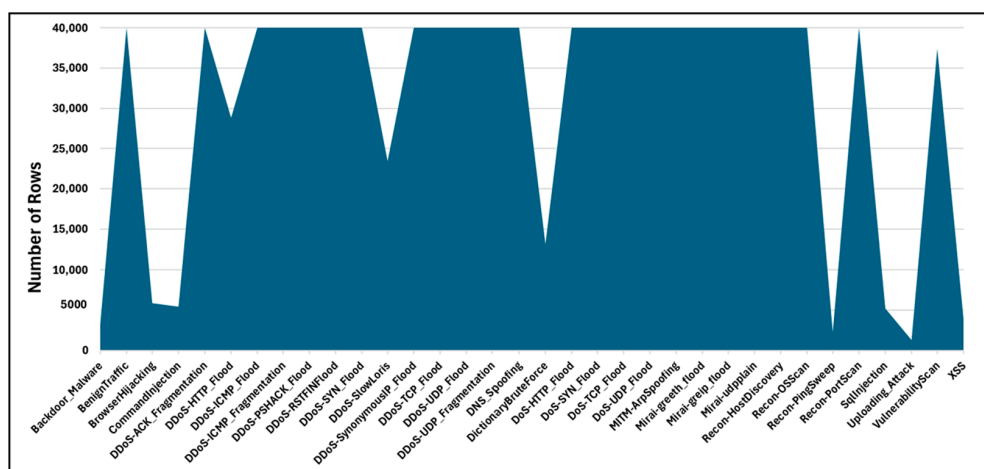


Figure 4. Randomly selected samples from original dataset.

3.3. Principal Component Analysis

PCA, a statistical technique for reducing dimensionality in high-dimensional datasets, transforms the original data into a new set of axes known as principal components (PCs). It does this while preserving the original data's most informative characteristics [42,43]. This ensures that despite the dimensionality reduction, the data remains rich in its informative content, instilling confidence in the data transformation process.

In the following, we will show the steps for computing the PCA in a mathematical concept [43].

The observed dataset X has n rows and p columns denoted by X_1, X_2, \dots, X_p , where n represents the number of observations, and p is the number of features. This dataset can be described as an $n \times p$ matrix. From this original dataset X , we can derive a $n \times p$ sample correlation matrix R . This correlation matrix captures the relationships between the p variables in the dataset.

The key step in principal component analysis (PCA) is to find the eigenvalues and eigenvectors of the correlation matrix R .

Let (λ_p, e_p) defines p as pairs of eigenvalues and eigenvectors of the matrix Z :

$$(\lambda_1, e_1), (\lambda_2, e_2), (\lambda_3, e_3), \dots, (\lambda_p, e_p) \quad (1)$$

PCA then uses these eigenvectors to transform the original dataset X into a new dataset Y , according to the following equation:

$$Y = R \cdot X \quad (2)$$

The PCs can be calculated as follows:

$$y_i = e_i^T (x - \bar{x}) \quad (3)$$

where $i = 1, 2, \dots, p$.

And the i th eigenvector can be calculated as follows:

$$e_i = (e_{i1}, e_{i2}, e_{i3}, \dots, e_{ip})^T \quad (4)$$

The sample mean of the observation x is defined as:

$$\bar{x} = (\sum_{i=1}^n x_i) / n \quad (5)$$

The eigenvalues are defined as follows:

$$Re = \lambda e \quad (6)$$

Assume that we have a vector of observations defined as follows:

$$Z = (z_1, z_2, z_3, \dots, z_p)' \quad (7)$$

$$z = (x - \bar{x}) \quad (8)$$

Equation (3) can be re-written as:

$$y_i = e_i^T z \quad (9)$$

where z is expressed as:

$$z = (x - \bar{x}) \quad (10)$$

The PCA possesses a few key characteristics summarized as follows [43]:

- The principal components (PCs) are uncorrelated.

- The first PC captures the highest variance in the data, followed by the second PC, and so on. All the PCs' combined variance equals the original variables' total variance. X_1, X_2, \dots, X_p .
- The PCs are arranged in descending order based on their corresponding eigenvalues, denoted as $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_p > 0$.

Note that the ratio of λ_i divided by the sum of all the eigenvalues characterizes the contribution of the i th PC to the overall variance of the data:

$$\lambda_i / \left(\sum_{i=1}^q \lambda_i \right) \quad (11)$$

Figure 5 shows the steps for performing PCA. First, data are fed into the covariance matrix after normalization. Normalization is a crucial step since the data have different scales. If the data are not normalized, biased results will arise. Second, the covariance matrix is computed from normalized data. Third, eigenvalue decomposition (EVD) on the covariance matrix is performed to factorize it into a set of eigenvectors and eigenvalues. Fourth, selecting the k -largest eigenvalues, which describe the highest variance in the data for a given direction. Finally, the principal component is computed by selecting only the eigenvectors with the k -largest eigenvalues, as the eigenvectors represent the principal directions of variation in the data. Note that the direction of the first PC is the direction in which the variance is maximum. The direction of the second PC is perpendicular to the first PC, in which the spread of the data is the largest. The third- and higher-order PCs are constructed similarly.

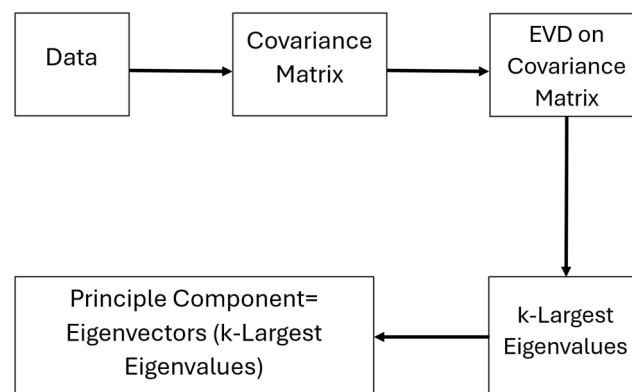


Figure 5. Procedure for computing PCs.

3.4. Monitoring of IoT Traffic Algorithms

The monitoring of IoT traffic comprises three algorithms. Algorithm 1 is used to transform the original data (IoT network traffic) into PCA components to reduce its dimensionality. Algorithm 2 is used to approximate the original data from the PCA components. Algorithm 3 sends the approximation of the original data to the ThingSpeak platform for monitoring the IoT traffic.

Algorithm 1 focuses on transforming IoT datasets into PCA components to reduce the dataset's dimensionality and is intended to run in the middleware layer. It takes the original dataset *OriginalData.csv* and the number of principal components *numComponents* as inputs to produce three outputs, which are the reduced dataset *PCA_Data.csv*, the *pcaVariance* object, and the data normalization object *scaler*. *pcaVariance* and *scaler* are saved to be used in Algorithm 2, especially when performing the inverse of PCA components as will be presented.

The procedure involves normalizing the features using the class from "sklearn.preprocessing" module, which is called *StandardScaler()* in the Scikit-learn library [44]. Then the PCA transformation is computed using the PCA function that is called from the "sklearn.decomposition" module to capture the largest variation of the data for predefined number of *numComponents*, which

creates a reduced dataset [45]. The algorithm then saves the reduced dataset, the *pcaVariance* object, and the normalization object *scaler*.

Algorithm 1: PCA Transformation

Input:

- 1: Load OriginalData.csv,
- 2: data = Read("OriginalData.csv"),
- 3: numComponents = 20

Output:

- 4: The PCA components data,
- 5: *pcaVariance*,
- 6: *scaler*

Procedure:

- 7: *scaler* = StandardScaler()
- 8: *normalizedFeatures* = *scaler.fit_transform*(data)
- 9: *pcaVariance* = PCA(numComponents)
- 10: *reducedFeatures* = *pcaVariance.fit_transform*(*normalizedFeatures*)
- 11: Save *reducedFeatures* to "PCA_Data.csv"
- 12: Save the *pcaVariance* and the *scaler* variables

End Algorithm

Algorithm 2 is used to approximate the original dataset by applying the inverse of PCA transformation. It is intended to be used in the cloud. Algorithm 2 takes the CSV file of PCA components, *PCA_Data.csv*, the previously saved *pcaVariance*, and *scaler* objects as inputs. The output of the algorithm is to approximate the original dataset. The procedure involves computing the inverse PCA transformation to approximate the original normalized features with the help of the function called "*pca.inverse_transform()*" from Scikit-learn library in Python. It then applies the normalization parameters to retrieve the original feature values. The algorithm saves this approximated dataset to a new CSV file called *approximateOriginalDataset.csv*.

Algorithm 2: PCA to Original

Input:

- 1: Load PCA_Data.csv,
- 2: data = Read("PCA_Data.csv"),
- 3: Load *pca*,
- 4: Load *scaler*,

Output:

- 5: Approximating the original data

Procedure:

- 6: *approxNormalizedFeatures* = *pca.inverse_transform*(data)
- 7: *approxOriginalFeatures* = *scaler.inverse_transform*(*approxNormalizedFeatures*)
- 8: Save *approxOriginalFeatures* to "approximateOriginalDataset.csv"

End Algorithm

Algorithm 3 transmits data to the ThingSpeak platform, which is used to monitor the IoT traffic. The initial step of the input parameter is to load the CSV file *approximateOriginalDataset.csv*, which was produced by Algorithm 2. Then the data of the CSV are read, and the data are stored in the variable *data*. After that, the configuration of communication with ThingSpeak channel is needed. This is achieved by assigning the unique values of the *channelID* and the *writeAPIKey* from the ThingSpeak account. The number of rows of the dataset is defined to be used later with a loop. The last parameter of the input is assigning the number of seconds to the *seconds* variable, which is used to pause the transmission to meet with ThingSpeak rate limit. The expected output of Algorithm 3 is to send data from the CSV file to the ThingSpeak platform. The procedure handles the sending of the data to

the corresponding fields of ThingSpeak, which is achieved by looping until each record is sent. Each field corresponds to a feature from the data, so the number of fields should match the number of features. In each iteration, one record of the data is transmitted every specific number of seconds to ThingSpeak.

Algorithm 3: Send to ThingSpeak

Input:

- 1: Load approximateOriginalDataset.csv,
- 2: data = read ("approximateOriginalDataset.csv"),
- 3: Set channelID,
- 4: Set writeAPIKey,
- 5: Set numRows,
- 6: Set seconds

Output:

- 7: Sending data to ThingSpeak

Procedure:

- 8: For each iteration (i = 1 to numRows)
- 9: field1 = data.Flow_duration(i)
- 10: field2 = data.Header_Length(i)
- 11: field3 = data.ProtocolType(i)
- 12:
- 13: field46 = data.Weight(i)
- 14: response =thingSpeakWrite(channelID,'Fields',[1,..,46],'Values',
- 15: [field1,..,field46,'WriteKey', writeAPIKey)
- 16:
- 17: pause(seconds)
- 18: End for

End Algorithm

4. Results and Discussion

This section demonstrates the experimental setup and discusses the research findings. It also shows the evaluation result to verify and validate the proposed framework.

The environment setup of this research is summarized in Table 2. It lists the tools used to conduct the results, the purpose of using each tool, and its specifications.

Table 2. Experimental setup.

Component Name	Purpose of Use	Specifications
MATLAB	It is used to preprocess the dataset and connect to ThingSpeak.	Version R2023b.
Python	It is used to compute PCA transformation, compute inverse PCA, and implement and evaluate NIDS.	PyCharm version 2023.3.2, Python version 3.8.
ThingSpeak	It is used to monitor and visualize the IoT data.	A platform.
Desktop Computer	It is used to perform the experiments.	GPU: NVIDIA® GeForce RTX™ 4090, Processor: Intel Core i9, RAM 32 GB, Hard disk 1TB SSD.

The proposed model is evaluated using a dataset named CIC-IoT Dataset 2023. The evaluation procedure of the proposed model is carried out in two main stages. Stage 1 discusses the assessment of approximating the original data after PCA transformation. In this stage, we compared the data reversed from the PCA transformation with the original dataset to measure the similarity between both datasets. The reason is to prove the effectiveness of the PCA transformation in reducing the dimension of transmitted data to the cloud for monitoring network traffic by the end users. Stage 2 discusses the

evaluation of a network intrusion detection system (NIDS) in detecting and classifying malicious activity in network traffic data that is transmitted to the cloud to alert the network administrator of the presence of a cyberattack. The NIDS is implemented using various models of machine learning and deep learning.

4.1. Stage 1: Evaluating PCA and Inverse PCA Transformation

The details of PCA transformation were discussed in the previous section. Initially, the dataset was transformed into PCA components, then the PCA components were reconstructed to approximate the original data. To measure the similarity of the original data and the approximated data, we used Pearson correlation coefficient. It is a statistical method to find the linear correlation between two variables. Let us assume that X is the first variable and Y is the second variable. To find Pearson correlation coefficient between the two variables, we can use Equation (12) [46], where r is the Pearson correlation coefficient, X_i is the sample point of the first data, and Y_i is the sample point of the second data. \bar{X} holds the mean of the first data, and \bar{Y} holds the mean of the second data.

$$r = \frac{\sum (X_i - \bar{X}) (Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}} \quad (12)$$

The value of r ranges from $[-1,1]$, -1 indicates a perfect negative correlation while 1 indicates a perfect positive correlation. Figure 6 lists the similarity of each feature in percentage between the original data and the approximated data after applying PCA transformation technique using Pearson correlation coefficient. X-axis represents each feature ordered by the similarity in percentage from high to low and the Y-axis represents the similarity in percentage. It is observed that 36 out of 46 features have a similarity percentage more than or equal to 90%, which shows a strong correlation between the two data. The overall average of the similarity between the two data is 94.43%.

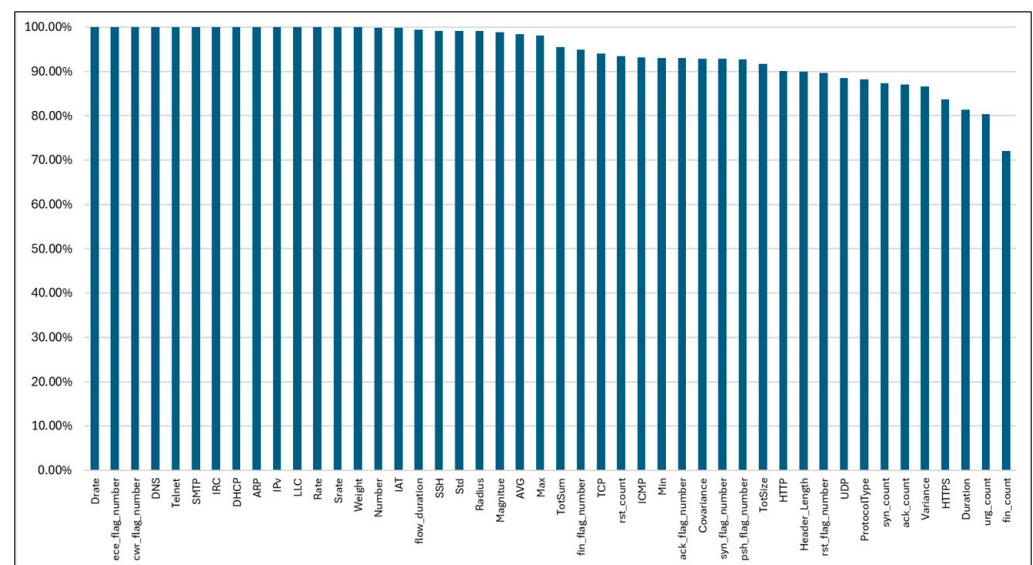


Figure 6. Similarity calculation.

4.2. Stage 2: Evaluation of NIDS

The NIDS is implemented based on ML models. We used various ML models, which are Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), 1D Convolutional Neural Networks (1D-CNN), Feedforward Neural Network (FFNN), Decision Tree (DT) and Random Forest (RF). Each model is evaluated based on classification metrics named test accuracy, precision, recall, and F1-score. Those metrics are shown in Equations (13)–(16), respectively [47]. Figure 7 illustrates the test accuracy of the ML models that are used

when predicting the attack type in IoT network traffic. The test accuracy is used to show the effectiveness of each machine learning in estimating the attack type by comparing the predicted data with the actual data. The test accuracy results range between 89.19% and 98.13%. The FFNN produces the lowest accuracy, and the LSTM produces the highest accuracy. LSTM and GRU are types of recurrent neural network (RNN) and both models score comparable accuracy results, they are effective with sequence learning and time series data [48,49]. 1D-CNN reports a test accuracy of 94.19%, and it uses convolutional filters. Additionally, it is effective in extracting features [50]. FFNN has the lowest accuracy of 89.19% and it is a basic model for neural networks. The DT is a simple model that predicts cyberattack types by splitting data into branches. The DT reaches an accuracy of 93.61%, however, RF has an accuracy slightly higher than DT, which is equal to 94.23%. The reason is that RF is an ensemble of decision trees, consisting of multiple trees that share the decision of the prediction of cyberattack type.

$$\text{Test Accuracy} = \left(\frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative}} \right) \times 100 \quad (13)$$

$$\text{Precision} = \left(\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \right) \times 100 \quad (14)$$

$$\text{Recall} = \left(\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \right) \times 100 \quad (15)$$

$$F_1 \text{ score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \times 100 \quad (16)$$

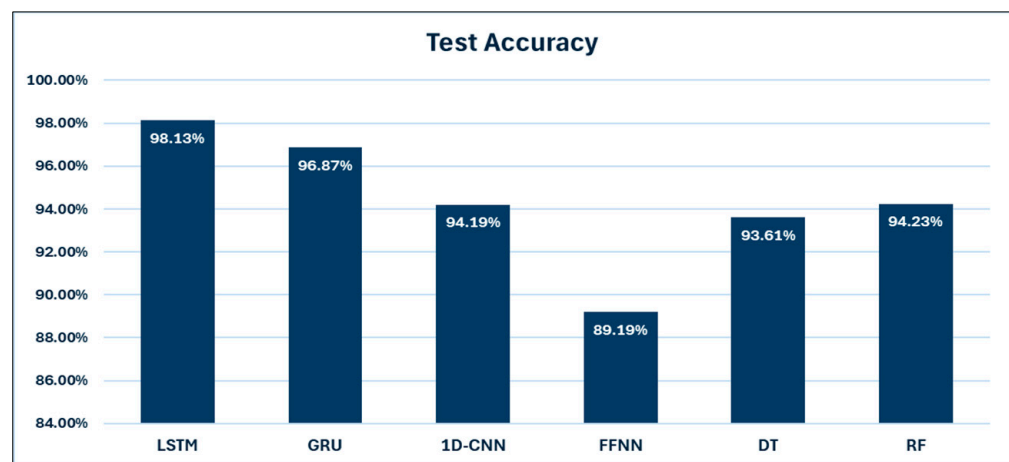


Figure 7. Test accuracy.

The macro average is used to calculate the average of precision, recall, and F1-score in Equation (17) [51]. The macro average treats each class equally when computing the average, which is preferred to be utilized with balanced data. Table 3 lists the macro average outcomes of different ML models to comprehensively compare them. It is essential to know the macro average is used to evaluate the overall performance of each ML in predicting cyberattack type with equal weight for each class. The LSTM model scores the highest results in each metric. This is reasonable because LSTM also has the highest test accuracy as discussed before. By contrast, FFNN has the lowest performance in this comparison. The GRU model approximates the performance of LSTM with precision, recall, and F1-score close to 95.70%. 1D-CNN demonstrates a decent performance, however, it is marginally less than the performance of LSTM and GRU models. The DT displays moderate performance

for precision, recall, and F1-score with 87.32%, 87.26%, and 87.24%, respectively. RF has a better precision value than DT, but it demonstrates lower recall and F1-score scores than DT.

$$\text{Macro Average} = \frac{\sum \text{metric}}{\text{Number of Classes}} \quad (17)$$

Table 3. Macro average of ML models.

Model #	Precision	Recall	F1-Score
LSTM	97.47%	97.29%	97.29%
GRU	95.71%	95.79%	95.79%
1D-CNN	91.53%	90.56%	90.15%
FFNN	78.82%	76.24%	76.53%
DT	87.32%	87.26%	87.24%
RF	91.79%	84.59%	86.82%

In addition, the weighted average for precision, recall, and F1-score is calculated using Equation (18) [52]. Initially, each metric is computed individually, then each metric value is weighted by the number of instances (support) for each class. After that, the sum of the weighted values is computed and finally, it is divided by the total number of supports. The weighted average is preferred to be utilized with imbalanced data. Table 4 presents the weighted average for precision, recall, and F1-score of different ML models. Overall, the results of Table 4 closely follow the results of Table 3 with slightly higher performance. According to the experiment result, LSTM demonstrates robust performance compared with other ML models; therefore, it can be selected when implementing NIDS for IoT traffic. In addition, GRU also performs a decent performance when predicting the cyberattack type. The reason for the high performance of LSTM and GRU is that they obtain good performance with time series data, which is a common characteristic of IoT traffic [53].

Table 4. Weighted average of ML models.

Model #	Precision	Recall	F1-Score
LSTM	98.23%	98.20%	98.14%
GRU	96.91%	96.83%	96.94%
1D-CNN	95.24%	94.25%	93.96%
FFNN	89.03%	89.27%	88.98%
DT	93.73%	93.68%	93.63%
RF	94.32%	94.39%	94.25%

In contrast, FFNN illustrates the lowest performance when predicting cyberattack type.

$$\text{Weighted Average} = \frac{\sum(\text{metric} \times \text{Support})}{\sum \text{Support}} \quad (18)$$

4.3. ThingSpeak Dashboard

An example of a ThingSpeak dashboard is illustrated in Figure 8. The dashboard can be used to visualize and observe each feature of the IoT traffic. We show only four features for discussion purposes. The dashboard allows the end user to monitor and analyze the streamed data of IoT traffic. As the figure shows, each feature can be visualized in a small plot that has an x -axis and a y -axis. The x -axis refers to the time and the y -axis refers to the value of the feature.

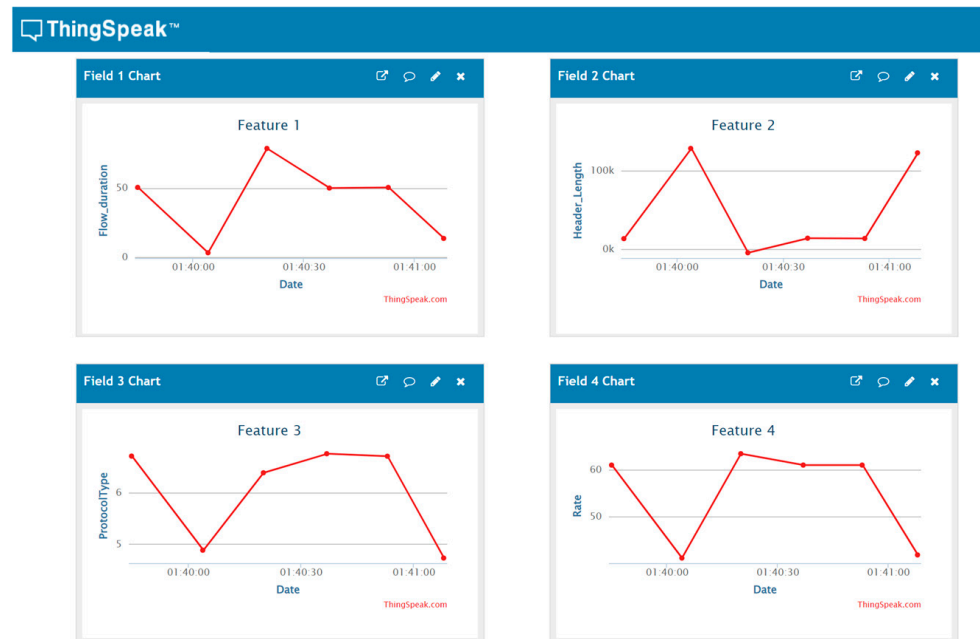


Figure 8. ThingSpeak dashboard.

5. Conclusions

In conclusion, this research proposed a comprehensive security framework for asymmetrical IoT architecture. It monitors network traffic and offers an NIDS to detect and classify cyberattacks. Network traffic is transmitted to the cloud after applying the PCA transformation technique to reduce the dimensionality of the data, which helps to send data to the cloud with less dimension. At the cloud, the PCA components are retransformed back to approximate the original data, therefore that data can be monitored by an end user to observe the stream traffic. Meanwhile, the NIDS, which was implemented based on ML performs classification to find malicious activity and report it to the cloud. The experiments of this research were evaluated to ensure their validity using two stages. In stage 1, we evaluated the performance of approximating the original data from the PCA components using the Pearson correlation coefficient. The overall average of the similarity between the approximated original data and the original data reached 94%, which demonstrated a strong correlation between the two data. In stage 2, we evaluated the performance of several well-known ML models using popular evaluation metrics such as test accuracy, precision, recall, and F1-score. Then we calculated the macro and weighted averages of each ML model and found out that LSTM produced superior performance, accomplishing 98% of test accuracy. Therefore, our security framework illustrated a promising result for monitoring IoT traffic and classifying cyberattack types. Our proposed security framework concentrates more on the computation aspect of performing the dimensionality reduction technique (PCA) of the data that are intended to be sent to the cloud with less complexity. In addition, it performs retransformation of the reduced data to approximate the original data (inverse PCA). In future work, we recommend researchers focus on applying our framework with the application side to assess its effectiveness in real-world scenarios while maintaining data integrity, security performance, and system scalability. In addition, since IoT traffic data have the characteristics of a time series, we recommend using different time series classification algorithms.

Author Contributions: Conceptualization, A.A., A.A.A. and B.A.; methodology, A.A. and A.A.A.; software, A.A., A.A.A. and B.A.; validation, N.A., B.A. and B.M.A.; formal analysis, N.A., B.A. and B.M.A.; investigation, A.A.; resources, A.A.A.; data curation, A.A. and A.A.A.; writing—original draft preparation, A.A., A.A.A. and B.A.; writing—review and editing, A.A., A.A.A., N.A., B.A. and

B.A.; visualization, B.M.A.; supervision, N.A.; project administration, A.A.A.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Graduate Studies and Scientific Research at Najran University under the Elite Funding Program grant code (NU/EP/SERC/13/288-2).

Data Availability Statement: The data presented in the study are contained in the article and openly available at <https://www.kaggle.com/datasets/akashdogra/cic-iot-2023> (accessed on 26 July 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Djenna, A.; Harous, S.; Saidouni, D.E. Internet of Things Meet Internet of Threats: New Concern Cyber Security Issues of Critical Cyber Infrastructure. *Appl. Sci.* **2021**, *11*, 4580. [CrossRef]
- Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Trans. Ind. Inf.* **2018**, *14*, 4724–4734. [CrossRef]
- Lionel Sujay Vailshery Number of Internet of Things (IoT) Connections Worldwide from 2022 to 2023, with Forecasts from 2024 to 2033. Available online: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/> (accessed on 27 July 2024).
- Lu, Y.; Xu, L. Da Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet Things J.* **2019**, *6*, 2103–2115. [CrossRef]
- Kumari, P.; Jain, A.K. A Comprehensive Study of DDoS Attacks over IoT Network and Their Countermeasures. *Comput. Secur.* **2023**, *127*, 103096. [CrossRef]
- Alam, T. Design a Blockchain-Based Middleware Layer in the Internet of Things Architecture. *JOIV Int. J. Inform. Vis.* **2020**, *4*, 28–31. [CrossRef]
- Khan, A.R.; Kashif, M.; Jhaveri, R.H.; Raut, R.; Saba, T.; Bahaj, S.A. Deep Learning for Intrusion Detection and Security of Internet of Things (IoT): Current Analysis, Challenges, and Possible Solutions. *Secur. Commun. Netw.* **2022**, *2022*, 4016073. [CrossRef]
- Ullah, I.; Mahmoud, Q.H. Design and Development of RNN Anomaly Detection Model for IoT Networks. *IEEE Access* **2022**, *10*, 62722–62750. [CrossRef]
- Vermesan, O.; Friess, P.; Guillemin, P.; Gusmeroli, S.; Sundmaeker, H.; Bassi, A.; Jubert, I.S.; Mazura, M.; Harrison, M.; Eisenhauer, M.; et al. Internet of Things Strategic Research Roadmap. In *Internet of Things—Global Technological and Societal Trends from Smart Environments and Spaces to Green Ict*; River Publishers: Aalborg, Denmark, 2011.
- Bajpai, S.; Sharma, K.; Chaurasia, B.K. Intrusion Detection Framework in IoT Networks. *SN Comput. Sci.* **2023**, *4*, 350. [CrossRef]
- Panahi Rizi, M.H.; Hosseini Seno, S.A. A Systematic Review of Technologies and Solutions to Improve Security and Privacy Protection of Citizens in the Smart City. *Internet Things* **2022**, *20*, 100584. [CrossRef]
- Javadpour, A.; Pinto, P.; Ja'fari, F.; Zhang, W. DMAIDPS: A Distributed Multi-Agent Intrusion Detection and Prevention System for Cloud IoT Environments. *Clust. Comput.* **2023**, *26*, 367–384. [CrossRef]
- Khraisat, A.; Alazab, A. A Critical Review of Intrusion Detection Systems in the Internet of Things: Techniques, Deployment Strategy, Validation Strategy, Attacks, Public Datasets and Challenges. *Cybersecurity* **2021**, *4*, 18. [CrossRef]
- Nisar, A. Intrusion Detection Systems: Categories, Attack Detection and Response. *SSRN Electron. J.* **2023**. [CrossRef]
- Yadav, N.; Pande, S.; Khamparia, A.; Gupta, D. Intrusion Detection System on IoT with 5G Network Using Deep Learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9304689. [CrossRef]
- Lone, A.N.; Mustajab, S.; Alam, M. A Comprehensive Study on Cybersecurity Challenges and Opportunities in the IoT World. *Secur. Priv.* **2023**, *6*, e318. [CrossRef]
- Karie, N.M.; Sahri, N.M.; Yang, W.; Valli, C.; KEBande, V.R. A Review of Security Standards and Frameworks for IoT-Based Smart Environments. *IEEE Access* **2021**, *9*, 121975–121995. [CrossRef]
- Husnain, M.; Hayat, K.; Cambiaso, E.; Fayyaz, U.U.; Mongelli, M.; Akram, H.; Ghazanfar Abbas, S.; Shah, G.A. Preventing MQTT Vulnerabilities Using IoT-Enabled Intrusion Detection System. *Sensors* **2022**, *22*, 567. [CrossRef]
- Zheng, Y.; Li, Z.; Xu, X.; Zhao, Q. Dynamic Defenses in Cyber Security: Techniques, Methods and Challenges. *Digit. Commun. Netw.* **2022**, *8*, 422–435. [CrossRef]
- Bai, L.; Hsu, C.; Harn, L.; Cui, J.; Zhao, Z. A Practical Lightweight Anonymous Authentication and Key Establishment Scheme for Resource-Asymmetric Smart Environments. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 3535–3545. [CrossRef]
- Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-Based Intrusion Detection System for IoT Networks through Deep Learning Model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
- Nickolaos Koroniotis, N.M. The Bot-IoT Dataset. Available online: <https://research.unsw.edu.au/projects/bot-iot-dataset> (accessed on 27 July 2024).
- Sampada Bhosale Network Intrusion Detection. Available online: <https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection> (accessed on 27 July 2024).
- Asgharzadeh, H.; Ghaffari, A.; Masdari, M.; Soleimani Gharehchopogh, F. Anomaly-Based Intrusion Detection System in the Internet of Things Using a Convolutional Neural Network and Multi-Objective Enhanced Capuchin Search Algorithm. *J. Parallel. Distrib. Comput.* **2023**, *175*, 1–21. [CrossRef]

25. Kasongo, S.M. A Deep Learning Technique for Intrusion Detection System Using a Recurrent Neural Networks Based Framework. *Comput. Commun.* **2023**, *199*, 113–125. [CrossRef]
26. Latif, S.; Huma, Z.E.; Jamal, S.S.; Ahmed, F.; Ahmad, J.; Zahid, A.; Dashtipour, K.; Aftab, M.U.; Ahmad, M.; Abbasi, Q.H. Intrusion Detection Framework for the Internet of Things Using a Dense Random Neural Network. *IEEE Trans. Ind. Inf.* **2022**, *18*, 6435–6444. [CrossRef]
27. Dina, A.S.; Siddique, A.B.; Manivannan, D. A Deep Learning Approach for Intrusion Detection in Internet of Things Using Focal Loss Function. *Internet Things* **2023**, *22*, 100699. [CrossRef]
28. Thakkar, A.; Lohiya, R. Attack Classification of Imbalanced Intrusion Data for IoT Network Using Ensemble-Learning-Based Deep Neural Network. *IEEE Internet Things J.* **2023**, *10*, 11888–11895. [CrossRef]
29. Altunay, H.C.; Albayrak, Z. A Hybrid CNN+LSTM-Based Intrusion Detection System for Industrial IoT Networks. *Eng. Sci. Technol. Int. J.* **2023**, *38*, 101322. [CrossRef]
30. Awajan, A. A Novel Deep Learning-Based Intrusion Detection System for IoT Networks. *Computers* **2023**, *12*, 34. [CrossRef]
31. Zhao, R.; Huang, Y.; Deng, X.; Shi, Y.; Li, J.; Huang, Z.; Wang, Y.; Xue, Z. A Novel Traffic Classifier With Attention Mechanism for Industrial Internet of Things. *IEEE Trans. Ind. Inf.* **2023**, *19*, 10799–10810. [CrossRef]
32. Alghamdi, R.; Bellaiche, M. An Ensemble Deep Learning Based IDS for IoT Using Lambda Architecture. *Cybersecurity* **2023**, *6*, 5. [CrossRef]
33. Abusitta, A.; de Carvalho, G.H.S.; Wahab, O.A.; Halabi, T.; Fung, B.C.M.; Mamoori, S. AI Deep Learning-Enabled Anomaly Detection for IoT Systems. *Internet Things* **2023**, *21*, 100656. [CrossRef]
34. Alrowais, F.; Althahabi, S.; Alotaibi, S.S.; Mohamed, A.; Ahmed Hamza, M.; Marzouk, R. Automated Machine Learning Enabled Cybersecurity Threat Detection in Internet of Things Environment. *Comput. Syst. Sci. Eng.* **2023**, *45*, 687–700. [CrossRef]
35. Yazdinejad, A.; Kazemi, M.; Parizi, R.M.; Dehghantanha, A.; Karimipour, H. An Ensemble Deep Learning Model for Cyber Threat Hunting in Industrial Internet of Things. *Digit. Commun. Netw.* **2023**, *9*, 101–110. [CrossRef]
36. Sharma, B.; Sharma, L.; Lal, C.; Roy, S. Anomaly Based Network Intrusion Detection for IoT Attacks Using Deep Learning Technique. *Comput. Electr. Eng.* **2023**, *107*, 108626. [CrossRef]
37. Almiani, M.; AbuGhazleh, A.; Al-Rahayfeh, A.; Atiewi, S.; Razaque, A. Deep Recurrent Neural Network for IoT Intrusion Detection System. *Simul. Model. Pract. Theory* **2020**, *101*, 102031. [CrossRef]
38. Jullian, O.; Otero, B.; Rodriguez, E.; Gutierrez, N.; Antona, H.; Canal, R. Deep-Learning Based Detection for Cyber-Attacks in IoT Networks: A Distributed Attack Detection Framework. *J. Netw. Syst. Manag.* **2023**, *31*, 33. [CrossRef]
39. ThingSpeak. Available online: <https://www.mathworks.com/products/thingspeak.html> (accessed on 12 August 2024).
40. Faruqui, N.; Yousuf, M.A.; Whaiduzzaman, M.; Azad, A.; Alyami, S.A.; Liò, P.; Kabir, M.A.; Moni, M.A. SafetyMed: A Novel IoMT Intrusion Detection System Using CNN-LSTM Hybridization. *Electronics* **2023**, *12*, 3541. [CrossRef]
41. Akash Dogra CIC IoT Dataset 2023. Available online: <https://www.kaggle.com/datasets/akashdogra/cic-iot-2023> (accessed on 26 July 2024).
42. Jolliffe, I. Principal Component Analysis. In *International Encyclopedia of Statistical Science*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1094–1096.
43. Hoang, D.H.; Nguyen, H.D. A PCA-Based Method for IoT Network Traffic Anomaly Detection. In Proceedings of the 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon, Republic of Korea, 11–14 February 2018; pp. 381–386.
44. StandardScaler. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> (accessed on 12 August 2024).
45. Sklearn. Decomposition. Available online: <https://scikit-learn.org/stable/api/sklearn.decomposition.html> (accessed on 12 August 2024).
46. Atoum, I. Scaled Pearson’s Correlation Coefficient for Evaluating Text Similarity Measures. *Mod. Appl. Sci.* **2019**, *13*, 10. [CrossRef]
47. Alsulami, A.A.; Abu Al-Haija, Q.; Alturki, B.; Alqahtani, A.; Binzagr, F.; Alghamdi, B.; Alsemmeiri, R.A. Exploring the Efficacy of GRU Model in Classifying the Signal to Noise Ratio of Microgrid Model. *Sci. Rep.* **2024**, *14*, 15591. [CrossRef]
48. TS, P.; Shrinivasacharya, P. Evaluating Neural Networks Using Bi-Directional LSTM for Network IDS (Intrusion Detection Systems) in Cyber Security. *Glob. Transit. Proc.* **2021**, *2*, 448–454. [CrossRef]
49. De Brouwer, E.; Simm, J.; Arany, A.; Moreau, Y. GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series. *arXiv* **2019**, arXiv:1905.12374.
50. Azizjon, M.; Jumabek, A.; Kim, W. 1D CNN Based Network Intrusion Detection with Normalization on Imbalanced Data. In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Fukuoka, Japan, 19–21 February 2020; pp. 218–224.
51. Gowda, T.; You, W.; Lignos, C.; May, J. Macro-Average: Rare Types Are Important Too. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021. [CrossRef]

52. Bonfietti, A.; Lombardi, M. The Weighted Average Constraint. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7514, pp. 191–206.
53. Khedkar, S.P.; Canessane, R.A.; Najafi, M.L. Prediction of Traffic Generated by IoT Devices Using Statistical Learning Time Series Algorithms. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5366222. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.