 doi:10.20944/preprints202210.0431.v1

*Article*

# Anomaly-based Intrusion Detection System for IoT Networks With Improved Data Engineering

**Abdulaziz A. Alsulami [1], Qasem Abu Al-Haija [2,*] and Ahmad Tayeb [3]**

[1] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; aalsulami10@kau.edu.sa
[2] Department of Cybersecurity, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan; q.abualhaija@psut.edu.jo
[3] Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia, ajtayeb@kau.edu.sa

* Correspondence: q.abualhaija@psut.edu.jo;

**Abstract:** Nowadays, the Internet of Things (IoT) devices and applications have rapidly expanded worldwide due to their benefits in improving the business environment, industrial environment, and people's daily lives. However, the IoT devices are not immune to malicious network traffic, which causes potential negative consequences and sabotages IoT operating devices. Therefore, developing a method for screening network traffic is necessary to detect and classify malicious activity to mitigate its negative impacts. Therefore, this research proposes a predictive machine learning model to detect and classify network activity in an IoT system. Specifically, our model distinguishes between normal and anomaly network activity. Furthermore, it classifies network traffic into five categories, normal, Mirai attack, denial of service (DoS) attack, Scan attack, and man-in-the-middle (MITM) attack. Five supervised learning models were implemented to characterize their performance in detecting and classifying network activities for IoT systems. This includes models shallow neural networks (SNN), decision trees (DT), bagging trees (BT), support vector machine (SVM), and k-nearest neighbor (kNN). The learning models were evaluated on a new and broad dataset for IoT attacks, the IoTID20 dataset. Besides, a deep feature engineering process was applied to the dataset to improve the accuracy of the learning models. Our experimental evaluation exhibited an accuracy of 100% recorded for the detection using all implemented models and an accuracy of 99.4%-99.9% recorded for the classification process.

**Keywords:** Supervised machine learning; intrusion detection; data engineering; cybersecurity; Internet of Things.

## 1. Introduction

Internet of Things (IoT) and Cyber-physical systems (CPS) technologies have considerably expanded our capability to realize our ecosystem and the surrounding world. IoT technology has touched almost every pitch of everyday life with its widespread applications. This, in turn, has substantially improved our life quality as a result of adopting the IoT "know-how" of several life, which have the potential to collect, harvest, and investigate data concerning the adjoining environment [1]. This context has accelerated the improvement of smart cities by enabling communication between things (machines) and between machines and humans. Such communications have recently been termed machine-to-machine (M2M) and machine-to-human (M2H) communication. IoT devices continue to expand swiftly and are being connected and spread through diverse applications and services. The number of IoT devices will likely exceed 125 billion by 2030 [2].

IoT system has been recently adopted in almost all areas of real-life applications. Many applications have been mentioned in the literature [3]. As such, smart cities require extensive use of technologies and connectivity resources to increase the overall quality of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

people's lives [4], smart environment involves multiple IoT applications like monitoring the snow level, fire detection, pollution monitoring, earthquakes, landslides, early detection [5], smart grids involve applications related to different monitoring, management, and measurements [6], smart agriculture which includes monitoring soil moisture, humidity, temperature, and selective irrigation in dry zones [7], home automation which contains various IoT applications such as remotely controlling electrical appliances to save energy, systems deployed (i.e., camera based on AI) on doors and windows disclosing intruders (hackers) [8], and security and emergencies include applications, for example, that allow only authorized persons to enter restricted (selected) areas and safe human and robotics interaction [9].

Even though IoT is a promising insightful technology with marvelous consequences and potential for spread and growth, IoT infrastructures are susceptible to various cyber-attacks and threats [10]. This is due to constrictions in processing capability, storage, memory capabilities, and communication capacity for the tiny energy-aware endpoint devices that reside within the IoT infrastructure. Indeed, confidentiality, integrity, and availability (CIA) are among the sizeable challenges of the IoT ecosystem [11]. Fig.1 illustrates the various cyber-attacks on IoT systems.

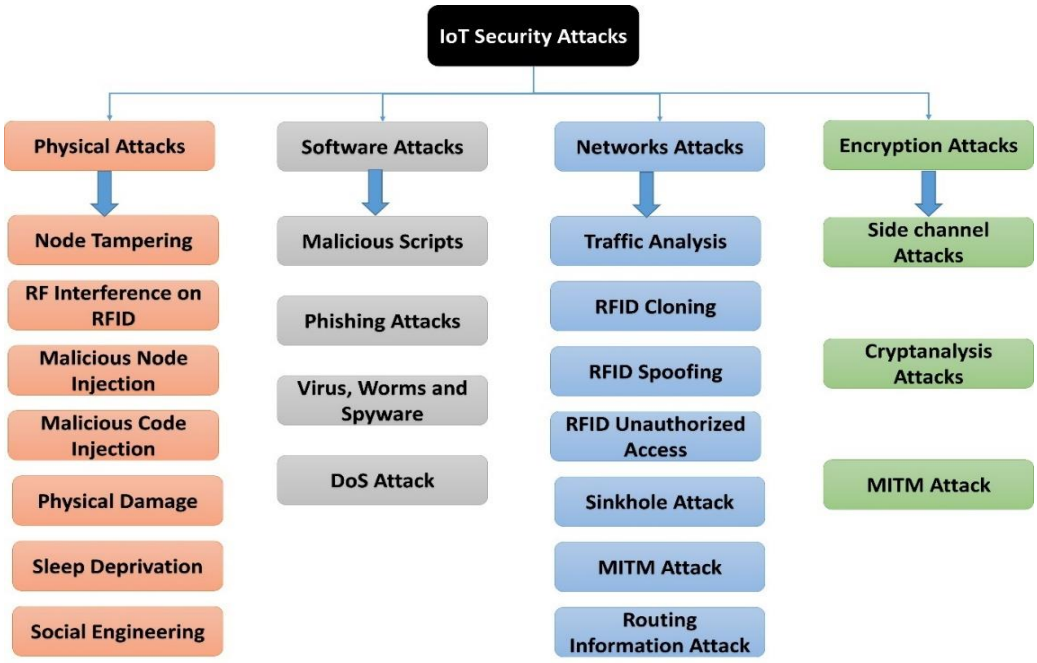


**Figure 1.** Main types of Cyber-attacks against the different layers of IoT systems

With the enormous and uninterrupted growth of cyber-attack occurrences in IoT infrastructures [12], it has become almost ridiculous to identify and thwart such attacks by means of conventional intrusion detection systems (IDSs) built based on the attack's signature. While the signature-based IDS can provide highly accurate and precise detection performance for the attacks/intrusions that match the pre-stored intrusion patterns (such as sequences of system calls, patterns of network traffic, … etc.), the problem occurs and even increases when a new attack (zero-day) is discovered. This is because traditional signature-based IDSs work depends on the pre-knowledge of a potential attack signature. Hence, they can detect an attack only if it is pre-deposited in their database.

Therefore, to tackle this limitation, an anomaly-based IDS has been proposed to replace the conventional IDS using adopting smarter and more intelligent techniques. Instead of matching the attack's signature with the pre-existing intrusion patterns, anomaly-based IDS defines a profile describing "normal" behavior and then detects deviations. This can detect potential new attacks (zero-day attacks). However, it still fails to detect all unknown attacks accurately in a dynamic environment such as an IoT ecosystem, and the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

cost of false detection rate is still high. Thus, many zero-day attacks remain undiscovered due to the existing limitations of IoT devices and conventional anomaly detection methods. Such functionalities are usually facilitated through vital and essential defense means such as a network intrusion detection system (NIDS), which examines network traffic for anomalous behaviors [13]. Fig.2 illustrates the typical deployment of NIDS in communication networks. To obtain a trusted environment and network, the anomaly-based-IDS can be utilized alongside conventional cyber-defense systems like firewall systems [14] to examine the network traffic, and anomaly-IDS can distinguish the traffic as benign or malicious by using its pre-trained models.

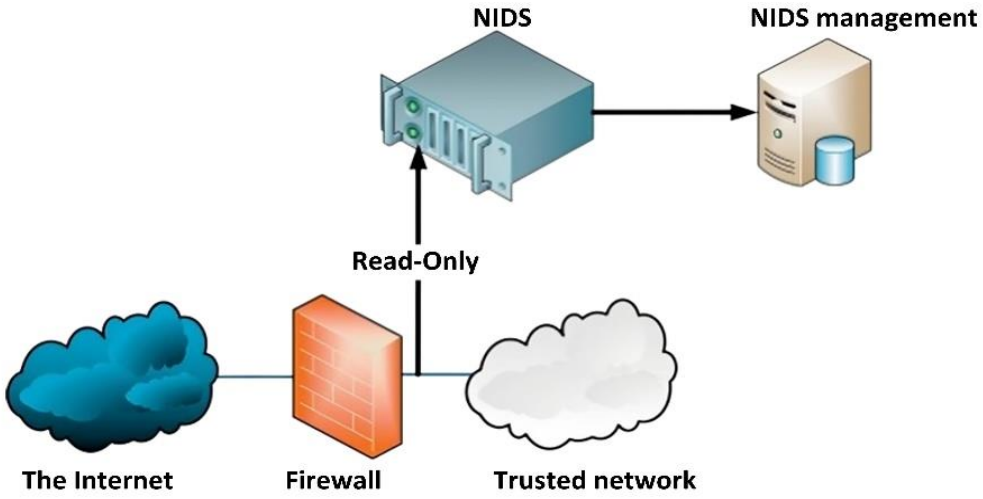


**Figure 2.** Typical NIDS architecture.

Consequently, over the past decade, big endeavors in handling security concerns related to intrusion/cyber-attacks detection in the IoT system. Most of these anomaly-based IDS systems were developed by employing the techniques of machine learning (ML) and deep learning (DL) techniques to provide intelligent cybersecurity decision-making. Since ML/DL techniques operate using datasets of records and features that are used to train and test the predictive IDS models, it should be noted that not all of the features/records in a dataset are relevant or significant while training/testing classification/detection models. Therefore, data engineering and feature preprocessing have formulated a core phase of every ML/DL-based IDS model that played a major role in making the raw data collected from the IoT ecosystem usable for further analysis and predictions. In anomaly detection challenges, for example, feature/data engineering is more significant in the IoT ecosystem since the features may include null or zero features. Relevant features, in some cases, are more difficult to extract by only ML/DL algorithms without using feature/data engineering approaches. Techniques of relevant features to identify attacks have been made to classify the data by industrial companies and researchers.

Several auspicious state-of-the-art models for anomaly intrusion detection models have been conducted for IoT cybersecurity using machine and deep learning approaches [15 - 31]. Tab.1 summarizes the reviewed research models for anomaly-based IDS using machine/deep learning approaches to solve cybersecurity concerns of cyber-attacks on IoT systems.

### *1.1 Our Contributions*

This study proposed an anomaly-based intrusion detection system that can detect the zero-day attacks of common IoT cyberattacks using machine learning techniques utilizing the sovereignty of Nvidia-Quad GPUs. Specifically, our model distinguishes between normal and anomaly network activity. Furthermore, it classifies network traffic into five categories, normal, Mirai attack, DoS attack, Scan attack, an MITM attack. Five supervised machine learning models, named Shallow Neural Networks (SNNs), Decision Trees

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

(DT), Bagged Tree (BT), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN), was implemented to detect and classify network activity in an IoT system. In addition, we have applied different data preprocessing and feature engineering processes to increase the prediction accuracy of the aforementioned machine learning models. As a result, the accuracy rates for all models have scored extremely high ratios rates between 99.40% to 100%. Such accuracy scores have outperformed the performance of all other existing models. Specifically, the main contributions of this paper can be summarized as follows:

- We present a comprehensive anomaly-based intrusion detection/classification system that can identify and classify the IoT traffic records of an IoTID20 dataset into two classes (normal and anomaly) or five classes (normal, Mirai attack, DoS attack, Scan attack, and MITM attack). We stipulate an illuminated depiction of our system modules and the machine learning algorithms.
- We provide an extensive feature engineering and data preprocessing framework that significantly improves the system performance evaluation. We provide a thorough development, validation environment, configurations, and extensive simulation results, to better perceive the proposed solution methodology. The system has been evaluated using standard performance indicators of machine learning models such as confusion matrix, accuracy, precision, recall, and F-score metric.
- We compare our findings with other related state-of-the-art works machine-learning-based intrusion detection systems (ML-IDSs) employing the same dataset. We show that our proposed system is superior.

**Table.1.** Summary of surveyed related research articles of supervised ML-based anomaly IDS.

| Ref. | Learning Models | Datasets | Number of Features/ Number of Records | Cyber-Attacks |
|---|---|---|---|---|
| [15] | Auto-Encoder, random forest (RF), naïve Bayes (NB), Linear/ Quadratic Discriminator | CICIDS2017 | 83 Features/ 2,830,540 records | Distributed DoS (DDoS), Heartbleed, structured query language (SQL) Injection, Botnet. |
| [16] | Particle Swarm (PSO), XG Boost, RF | IoTID20 | 83 Features/ 450,00 records | Mirai, DoS, Scan, MITM |
| [17] | Auto-Encoders (AEs) | NSL-KDD/ IoTID20/ N-BaIoT | 43 Features/140,000 83 Features/450,000 114Features/612,000 | Norm,DoS, Probe, R2L, U2R / Mirai, DoS, Scan, MITM / Normal, Bashlite, Mirai |
| [18] | Convolutional neural network (CNN), long short-term memory (LSTM), CNN-LSTM | NSL-KDD/ IoTID20/ | 43 Features/140,000 83 Features/450,000 | Norm, DoS, Probe, root to local (R2L), user to root (U2R), / Mirai, DoS, Scan, MITM |
| [19] | LightGBM, Optimized Adaptive and Sliding Windowing (OASW) | NSL-KDD/ IoTID20/ | 43 Features/140,000 83 Features/450,000 | Norm,DoS, Probe, R2L, U2R / Mirai, DoS, Scan, MITM |
| [20] | Shallow CNN | NSL-KDD | 43 Features/ 150,000 Records | Norm,DoS, Probe, R2L, U2R |
| [22] | Bagging, J48, KNN, Multilayer Perceptron (MLP), Ensemble. | NSL-KDD/ IoTID20 | 11-60 Features/ 150,00−450,00 | Norm,DoS, Probe, R2L, U2R / Mirai, DoS, Scan, MITM |

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

| | | | | |
|---|---|---|---|---|
| [23] | Adaboost, DT | KDDCUP99, UNSW-NB15, NSL-KDD, CI-CIDS2017 | 43-100 Features/ 140,000– 612,000 | DDoS, flooding, U2R, Jamming |
| [24] | Gradient Boosting Machines, RF, NB, Deep Neural Network (DNN) | ToN_IoT | 7 Features/ 1,300,000 records | Normal, DoS, DDoS, Injection, MITM, Password, Scan, Cross-site scripting (XSS), Backdoor, Ransome. |
| [25] | SVM, NB, SNN, RF | N_BaIoT, Bot_IoT | 114 Features/ 612,000 records | Normal, Bashlite, Mirai |
| [26] | Adaboost, RusBoost, Bagging, Ensemble | WUSTL_IIOT-2018, N_BaIoT, and Bot_IoT | 100-114 Features/ 100,000-612,000 | Normal, Bashlite, Mirai, Port/Address Scanner. |
| [27] | AdaBoost | CICIDS 2019. | 88 Features/ 4,201,795 Records | DDoS, Heartbleed, SQL Injection, Botnet. |
| [28] | SNN, SVM, NB, RF, Self-organizing map | NSL-KDD, KDDCup99, ADFA-LD12, UNSWNB15 | 43-100 Features/ 140,000– 612,000 | DDoS, flooding, U2R, Jamming |
| [29] | Ensembles:(Boosted DT, Subspace kNN, RUSBoosted DT), SNN, Bilayered NN, Logistic Regression Kernel | Distilled-Kitsune-2018/ NSL-KDD dataset | 43 Features/ 145,00–150,000 | Mirai, operating system (OS) Scan, Fuzzing, Video Injection, Address Resolution Protocol (ARP), Wiretap, simple service discovery protocol (SSDP), Synchronous DoS, secure sockets layer(SSL)/DoS, Probe, R2L, U2R |
| [30] | Beta Mixture Model | BoT-IoT 21 | 12 Features / 3,000,000 records | DDoS, DoS, OS and Service Scan, Keylogging, and Data exfiltration attacks |
| [31] | AdaBoost DT | TON_IoT_2020 datasets | 7 Features/ 1,300,000 | DoS, DDoS, Injection Attacks, MITM, Password Attacks, Scanning, XSS Attacks, Backdoor attacks, and Ransomware attacks. |

### *1.2 Background of Machine Learning*

Machine learning is the main stage of this research because it is used to detect and classify network activity attacks, as was mentioned above. Varieties of supervised machine learning classifiers were being used: Shallow Neural Networks (SNNs), Decision Trees (DT), Bagged Trees (BT), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN).

Shallow Neural Networks (SNNs) is a feedforward neural networks that use multilayer perceptron (MLP) [32]. SNNs can be used to solve classification and regression problems based on supervised learning. Two SNNs models were developed; the first model predicts two classifications (label feature), and the second predicts five classifications

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

(category feature). Fig.3 depicts the second model. The input layer contains 71 input nodes, the hidden layer has ten nodes, and the output layer has five. The 71 features from the dataset were fed to the SNNs model and then processed by ten hidden nodes. Finally, the model predicts the five categories. For the detection procedure, we have a similar SNN model; however, the model has two output nodes instead of five nodes.

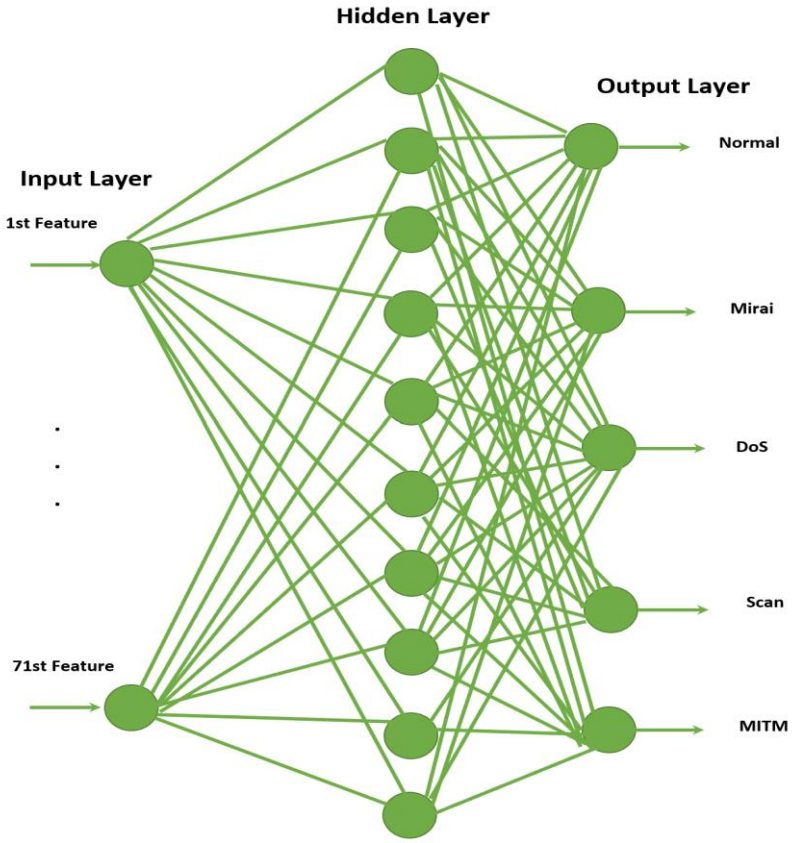


**Figure 3.** Shallow Neural Networks

Decision Trees (DT) is a widely used machine learning method in various fields such as image processing, pattern recognition, and classification [33]. DT can handle a vast dataset size. Two Decision Trees models were developed, the first model to predict the label feature and the other to predict the category feature. In addition, two Bagged Trees (BT) models were developed for classification purposes, one to predict two classifications and another to predict five classifications [34].

Support Vector Machine (SVM) is a powerful technique for solving classification and regression and linear and nonlinear problems [35]. The dataset is classified based on hyperplanes (lines). Two SVM models were developed, the first for predicting the label feature and the second for predicting the category feature. K-Nearest Neighbour (KNN) is a machine learning method that can be used as a classifier [36]. It classifies dataset points based on similarity; therefore, data points with similarities are close to each other. Two KNN models were developed: the first for predicting the label feature and the second for predicting the category feature. Fig.4 shows the training and classification process of DT, BT, SVM, and KNN models.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

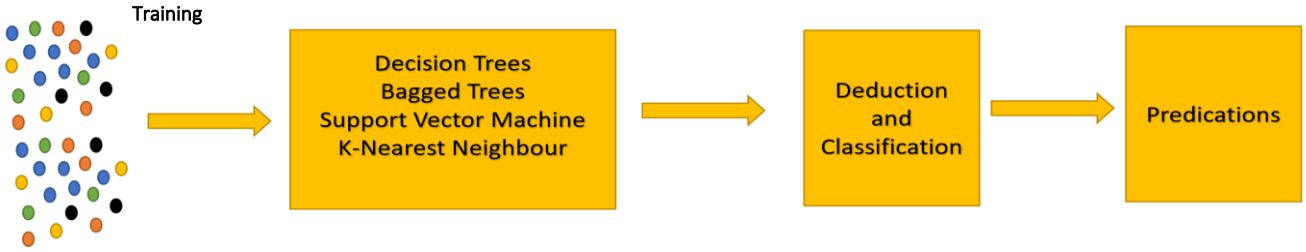


**Figure 4.** Machine Learning Classifiers

The rest of this paper is organized as follows: Dataset collection and data engineering are discussed and elaborated on in this section. It also introduces and justifies the dataset of IoT cyber-attacks employed by our system. Section 3 provides details of the proposed system architecture, development, data preprocessing, and detailed design steps. Section 4 presents the simulation environment for system implementation, testing, and validation and discusses the details of experimental evaluation, comparison, and discussion. Finally, Section 5 concludes the findings of the research.

## 2. Data Collection and Engineering

This section discusses the dataset used in this research to evaluate the anomaly-based IDS for the IoT system and the data engineering performed over the dataset to improve the learning and validation processes.

### *2.1 Dataset of IoT System*

IoT devices can operate in many domains, such as smart cities, healthcare, education, smart homes, smart grids, and transportation systems [37]. Our research concentrates on a smart home IoT system; therefore, the IoTID20 dataset [38] was used to test the performance of our model. The environment used to collect the IoTID20 dataset consists of IoT devices connected through an access point network [38]. The IoT devices comprise a laptop and a smartphone to establish intrusion attacks. The security camera and the AI speaker are the victims, as shown in Fig.5. A detail about the experiment can be found in this reference [38].

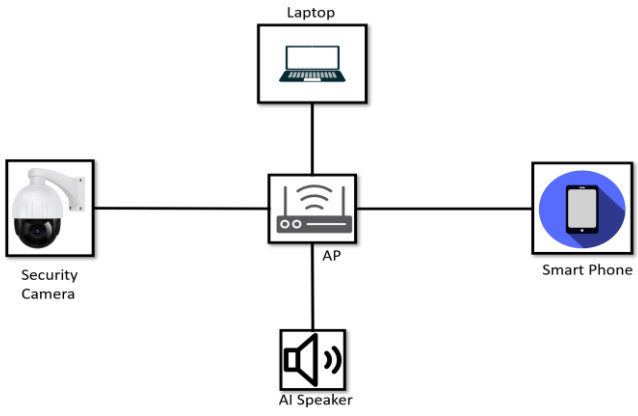


**Figure 5.** IoT devises Architecture

The dataset was collected from a real-time scenario using IoT devices, as shown in Fig.5. The original IoTID20 dataset includes 86 columns and 625,783 rows. Each row in the dataset is labeled with the type of network activity [39]. We preprocessed the dataset to increase the classification accuracy of the label, category, and sub-category features. However, we focused our study on label and category features and will express the reason in the Features Engineering section. Label features include binary classification, which is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

normal, and anomaly. Category features have five classifications: normal, Mirai attack, DoS attack, Scan attack, and MITM attack [40].

### *2.2 Features Engineering*

Features engineering is removing unnecessary features or extracting new features from existing features to increase the accuracy of the machine learning models [41].

Duplicated records were removed from the original dataset, and there were 164,087 duplications of records. As a result, the dataset becomes having 461,696 records. Tab.2 represents statistical information about the dataset used in this research. Moreover, the dataset has the source IP address (Src_IP) and destination IP address (Dst_IP) as features. However, machine learning models cannot sufficiently handle the format of IP addresses, such as "192.168.0.13" [42]. Therefore, to solve this issue and help the machine learning models obtain the most use of IP address information, we split the four IP address parts, octet numbers, into features, e.g., Src_IP_oct1: 192 Src_IP_oct2: 168, Src_IP_oct3: 0, and Src_IP_oct4: 13. By doing so, the machine learning model can understand and distinguish between the network and host portions. Furthermore, the IoTID20 dataset has a timestamp as a feature. Therefore, we extract the following information from the timestamp feature and include them in the dataset as new features: day of the week, hour, and am or pm to use it more efficiently. According to our experiment, those new features helped increase detection accuracy and machine learning classification. Finally, we converted the label and category string values to numerical values. For example, we map the values of the label feature normal to 0 and anomaly to 1, as shown while the numerical conversion of the category feature was normal (0), Mirai (1), DoS (2), Scan (3), and MITM (4).

**Table 2.** IoTID20 Dataset Statistics.

| Lable | Number of Records | Category | |
|---|---|---|---|
| Normal | 38598 | Normal | 38598 |
| Anomaly | 423098 | Mirai | 281102 |
| | | DoS | 59390 |
| | | Scan | 56744 |
| | | MITM | 25862 |

The Minimum Redundancy and Maximum Relevance (MRMR) algorithm are used for the feature selection procedure [43]. Each feature is ranked based on minimum redundancy and maximum relevance and assigned an importance score [44]. Therefore, a feature with a high score is more important than a less-score feature. In addition, a large drop in the rank between features will ease the feature selection. However, a small drop will make the feature selection more challenging. Thus, this research discarded the sub-category feature because after performing the MRMR algorithm on the sub-category feature, we observed that the drop in score between the 11th and the 72nd is relatively small, as shown in Fig.6.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

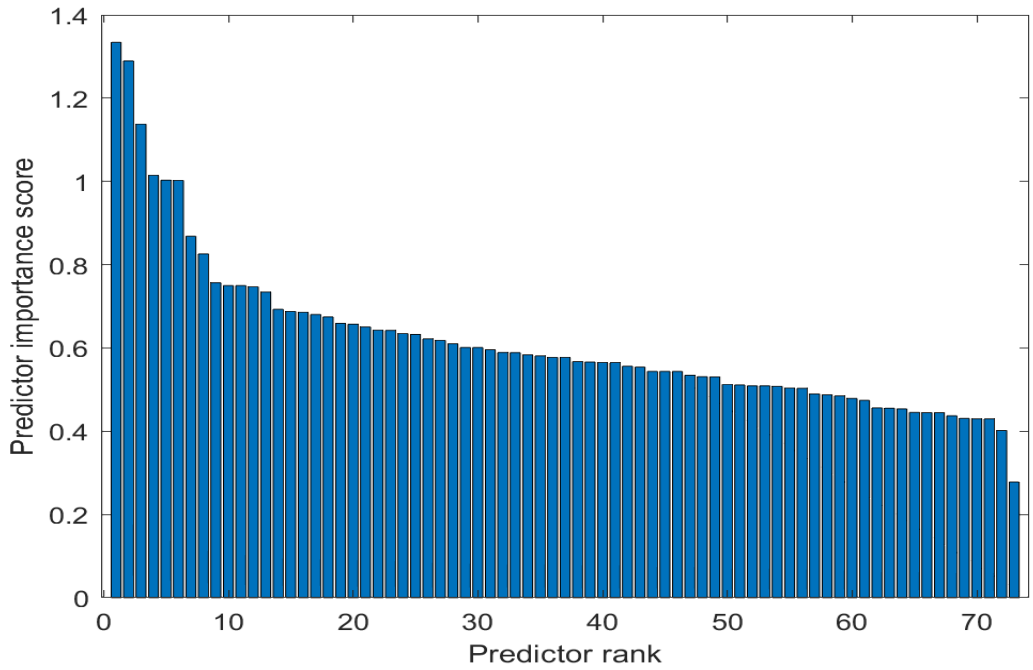


**Figure 6.** MRMR Algorithm

By looking at Fig.7, we can observe that intrusion attacks occurred every day of the week except Monday. In addition, most of the intrusion attacks take place on Thursday. Fig.8 illustrates whether the network traffic occurred in the morning or evening. It is worth saying that most of the network traffic recorded in the morning was intrusion attacks, and a few traffic was normal network packets. However, a few network traffic occurred in the evening and were intrusion activities.

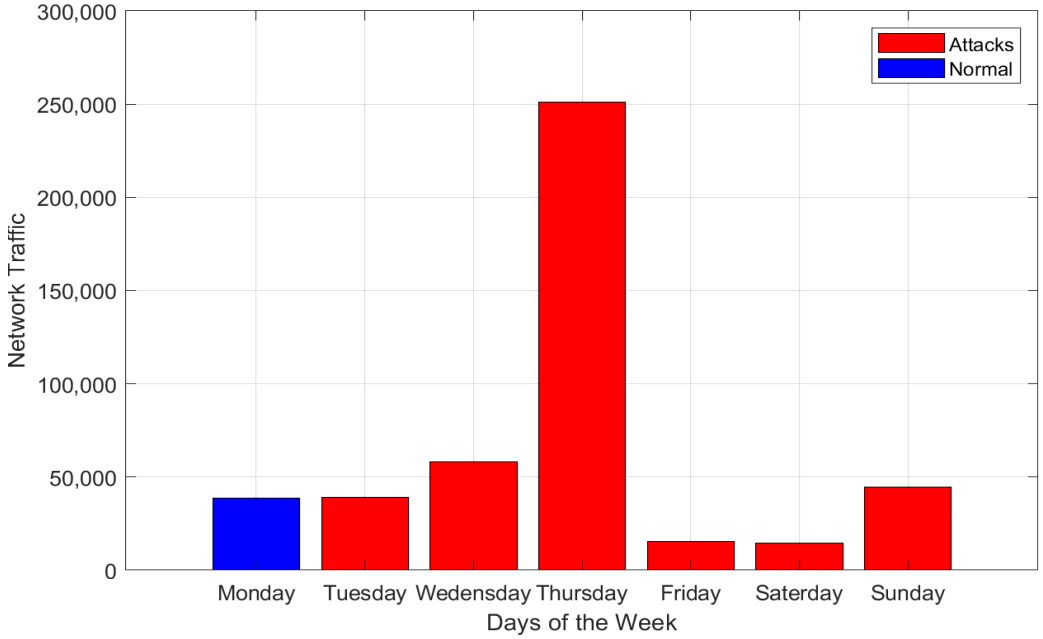


**Figure 7.** Network Traffic During the week

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

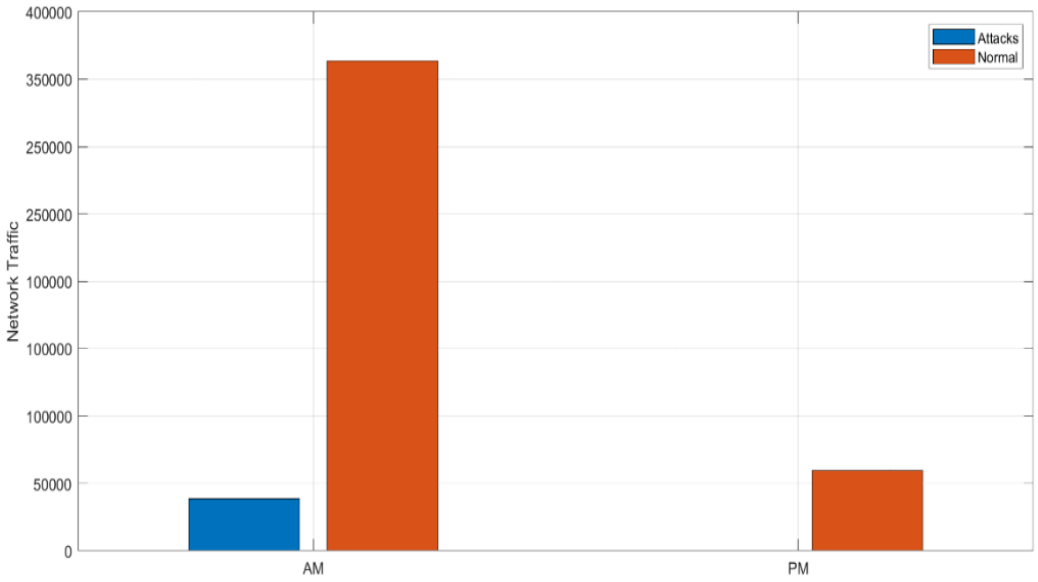


**Figure 8.** Network Traffic in Morning and Evening

Fig.9, part A shows features excluded from the dataset because their values are mostly zeros. However, Fig.9-part B offers features that were used in this study which are 74 features.

**(A) Removed Features**

Flow_ID, Src_IP, Dst_IP, Timestamp, Fwd_PSH_Flags, Fwd_URG_Flags, Fwd_Byts/b_Avg, Fwd_Pkts/b_Avg, Fwd_Blk_Rate_Avg, Bwd_Byts/b_Avg, Bwd_Pkts/b_Avg, Bwd_Blk_Rate_Avg, Init_Fwd_Win_Byts, Fwd_Seq_Size_Min, Bwd_PSH_Flags, Bwd_URG_Flags, FIN_Flag_Cnt, SYN_Flag_Cnt, RST_Flag_Cnt, URG_Flag_Cnt, CWE_Flag_Count, and ECE_Flag_Cnt.

**(B) Used Features**

Src_Port, Dst_Port, Protocol, Flow_Duration, Tot_Fwd_Pkts, Tot_Bwd_Pkts, TotLen_Fwd_Pkts, TotLen_Bwd_Pkts, Fwd_Pkt_Len_Max, Fwd_Pkt_Len_Min, Fwd_Pkt_Len_Mean, Fwd_Pkt_Len_Std, Bwd_Pkt_Len_MaxBwd_Pkt_Len_Min,Bwd_Pkt_Len_Mean, Bwd_Pkt_Len_Std, Flow_Byts/s, Flow_Pkts/s, Flow_IAT_Mean, Flow_IAT_Std, Flow_IAT_Max, Flow_IAT_Min, Fwd_IAT_Tot, Fwd_IAT_Mean, Fwd_IAT_Std, Fwd_IAT_Max, Fwd_IAT_Min, Bwd_IAT_Tot, Bwd_IAT_Mean, Bwd_IAT_Std, Bwd_IAT_Max, Bwd_IAT_Min, Fwd_Header_Len, Bwd_Header_Len, Fwd_Pkts/s, Bwd_Pkts/s, Pkt_Len_Min, Pkt_Len_Max, Pkt_Len_Mean, Pkt_Len_Std, Pkt_Len_Var, ACK_Flag_Cnt, Down/Up_Ratio, Pkt_Size_Avg, Fwd_Seg_Size_Avg, Bwd_Seg_Size_Avg,Subflow_Fwd_Pkts, Subflow_Fwd_Byts, Subflow_Bwd_Pkts, Subflow_Bwd_Byts, Init_Bwd_Win_Byts, Fwd_Act_Data_Pkts, Active_Mean, Active_Std, Active_Max, Active_Min, Idle_Mean, Idle_Std, Idle_Max, Idle_Min, Src_IP_oct1, Src_IP_oct2, Src_IP_oct3, Src_IP_oct4, Dst_IP_oct1, Dst_IP_oct2, Dst_IP_oct3, Dst_IP_oct4, Timestamp_DayOfWeek, Timestamp_Hour, Timestamp_AmPm_n, Label_n, Cat_n, Sub_Cat_n

**Figure 9.** Removed and Used Features.\

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

## 3 System Development and Specifications

This section discusses the data models and preprocessing used in this research by explaining the IoT system's architecture and a detailed description of the development and implementation of machine learning models used for detection and classification. Finally, it discusses the conducted simulation experiments, training, testing, and validation of the results. Classification is an intelligent technique to place a particular data set into a specific category based on predefined criteria [39]. In our case, the machine learning models are supposed to detect and classify IoT intrusion attacks by prediction procedure based on 74 selected features. The detection and classification machine learning models used in this research are supervised learning, so the models estimate the target output based on the chosen features [40]. This paper used machine learning models to predict the label and category features of the IoTID20 dataset. The architecture of the system model used in this research is illustrated in Fig.10.

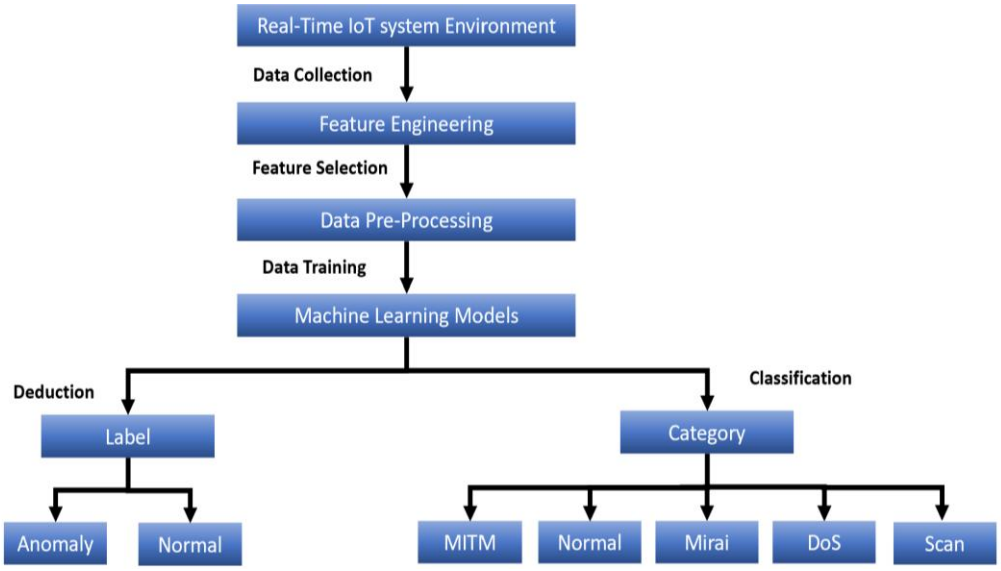


**Figure 10.** The architecture of ML models for prediction instruction attacks

### *3.1 Data Preprocessing*

Data preprocessing is a technique to prepare the dataset to be fed to a machine learning model [45]. Fig.11 depicts the preprocessing step. Initially, the dataset was stored in a Comma-Separated Value (CSV) format. Next, any string value of the matrix is converted to a numerical record, as was discussed in the Feature Engineering section. Then, the CSV file is converted to a MAT file (Matlab matrix). After that, the dataset was normalized, so each matrix value had a value between 0 and 1. For the data partitioning procedure, data is randomly divided into parts 70 % for training, 25% for testing, and 5 % for validation. We used across-validation technic as a validation scheme for our research. Finally, data is fed to the machine learning model, which will be discussed next.

### *3.2 Detection and Classification Procedures*

The detection procedure generates the label feature, which consists of two classifications, and the output is either normal or anomaly using the machine learning models mentioned earlier. The classification procedure generates the category feature, which consists of five classifications. The output is either normal, Mirai attack, DoS attack, Scan attack, or MITM attack using the earlier machine learning models.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

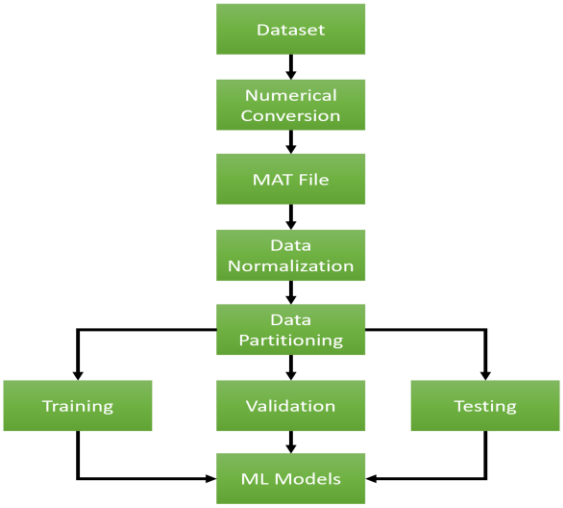


**Figure 11.** Data Pre-processing

### *3.3 Implementation and Validation Environment*

The IoTID20 dataset was used to train, validate and test our proposed detection and classification models. The aforementioned machine learning classifiers (i.e., SNN, DT, NB, SVM, and KNN) were trained, tested, and validated using the IoTID20 dataset. MATLAB® version 2022a was used to develop, test, and validate the five machine learning classifiers. Tab.3 briefly describes the hardware and software environment the authors used to experiment.

**Table 3.** Hardware and Software Description.

| Hardware / Software | Description |
|---|---|
| MATLAB | Version 2022a |
| CPU | Intel® Core™ i7-9750H CPU @ 2.60 GHz |
| Memory | 16.0 GB |
| GPU | NVIDIA GeForce RTX 2070 GDDR6 @ 8 GB |

## 4. Results and Discussion

This research proposes predictive models based on machine learning to detect and classify network activity. Ten models were trained, tested, and validated, five for detection and the remaining for classification purposes. For the detection model, network activities are classified into two groups (normal and anomaly). Meanwhile, for the classification model, network activities are classified into five groups (Normal, Mirai attack, DoS attack, Scan attack, and MITM attack)

### *4.1 Accuracy Evaluation*

We evaluated our machine learning models using the confusion matrix shown in Fig.12, which depends on True Positive Rate (TPR) and False Negative Rate (FPR). First, TPR and FPR are calculated using Eq. (1) and Eq. (2), respectively. Then the accuracy is calculated using Eq. (3) [46].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

| | | Real Label | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted Label | Positive | True Positive (TP) | False Positive (FP) |
| | Negative | False Negative (FN) | True Negative (TN) |

**Figure 12.** Two Class Confusion Matrix for calculation of the TPR, FPR.

$$TPR = TP/(TP + FN) \quad (1)$$

$$FPR = FP/(FP + TN) \quad (2)$$

$$Accuarcy = (TP + TN)/(TP + TN + FP + FN) \quad (3)$$

TP refers to the number of positive data classified correctly, and FN refers to the number of positive misclassified data. Likewise, FP refers to the number of negative misclassified data, and TN refers to the number of negative data classified correctly. In addition, we have evaluated our models in terms of other standard metrics, including precision, recall, and F1-Score, as represented in Tab.4.

By looking at Tab.4, it can be observed that our detection ML models achieved 100 % accuracy in the three metrics (Precision, recall, and F1- Score). In addition, we accomplished between 99.12% to 99.99% for classification ML models. The reason is that the comprehensive and enhanced data engineering as we have thoroughly investigated the dataset to come up with optimal (best) features that led to almost optimal performance of (precision, recall, and F1) in the case of detection ML models. We discussed the data engineering process in section 2.2

$$Precision = TP/(TP + FP) \quad (4)$$

$$Recall = TP/(TP + FN) \quad (5)$$

$$F1 - Score = 2 * (Precision * Recall)/(Precision + Recall) \quad (6)$$

**Table 4.** Accuracy Evaluation Results.

| ML Model | Detection / Classification | Precision | Recall | F1-Score |
|---|---|---|---|---|
| SSNs | Detection | 100% | 100% | 100% |
| SSNs | Classification | 100% | 99.99% | 99.99% |
| DT | Detection | 100% | 100% | 100% |
| DT | Classification | 99.99% | 99.99% | 99.99% |
| BT | Detection | 100% | 100% | 100% |
| BT | Classification | 99.99% | 99.99% | 99.99% |
| SVM | Detection | 100% | 100% | 100% |
| SVM | Classification | 99.81% | 99.78% | 99.79% |
| KNN | Detection | 100% | 100% | 100% |
| KNN | Classification | 99.36% | 99.88% | 99.12% |

The confusion matrices for the detection model (binary classification) of all ML techniques were equal for all and are shown in Fig.13(a); thus, no need to be repeated.

### *4.2 Our Results*

Fig.13(b) illustrates the confusion matrix of the classification model using SNNs. In the case of the detection model, our SNNs have no miss-labeled traffic, and the total network traffic classified as normal is 38596. However, 423096 of the traffic is classified as anomaly traffic. In the case of the classification model, only twelve traffic of the total network traffic was miss-labeled. Therefore, the overall accuracy of the two models reached 100%.

The performance of DT is shown in Fig.13(c), which shows the confusion matrix of the DT classification model. There is no miss-classified traffic in the detection model, and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

only thirteen network activities were miss classified in the classification model. Therefore, the overall accuracy of the two models reached 100%.

The evaluation performance of BT is represented in Fig.13(d), which shows the confusion matrix of the BT classification model. No miss-classified traffic using the detection model, and only sixteen traffic were miss classified using the classification model. In brief, the accuracy of the detection and classification models is 100%.

The performance response of SVM is shown in Fig.13(e), which illustrates the confusion matrixes of the SVM classification model. There was no miss classified traffic using the detection model, and only 487 out of 461,696 network activities were miss classified using the classification model. In summary, the accuracy of the detection model is 100%, and the overall accuracy of the classification model reached 99.80%.

The performance of KNN models is shown in Fig.13(b), which illustrates the confusion matrix of the KNN classification model. Again, the accuracy of the detection model is 100%, and the overall accuracy of the classification model reached 99.40%.

### *4.3 Comparing our Findings with Existing Results*

To our best knowledge, Tab.5 lists the recent machine models researchers developed to detect or classify the IoTID20 dataset. The table lists two types of classification used by researchers: detection (binary classification) and classification (multiclass classification). For machine learning, it is generally simpler to perform binary classification than multiclass classification [47]. The reason is that in binary classification, the ML needs to select from two decisions, i.e., 0 or 1; however, with multiclass classification, ML needs to choose from more than two decisions and perform sub-binary classification.

We can observe that our results slightly exceed other results. Also, it is worth saying that in this research [38], the authors used several machine learning classifiers such as DT, SVM, and Ensemble to detect and classify network activities in the IoTID20 dataset. They claimed they reached %100 using DT for detection and classification models. However, they accomplished low accuracy using SVM (less than 80% in the detection model and less than 50% in the case of the classification model); however, we reached the accuracy of 100% for the detection model and 99.80% for the classification model using SVM due to the feature engineering we discussed earlier.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

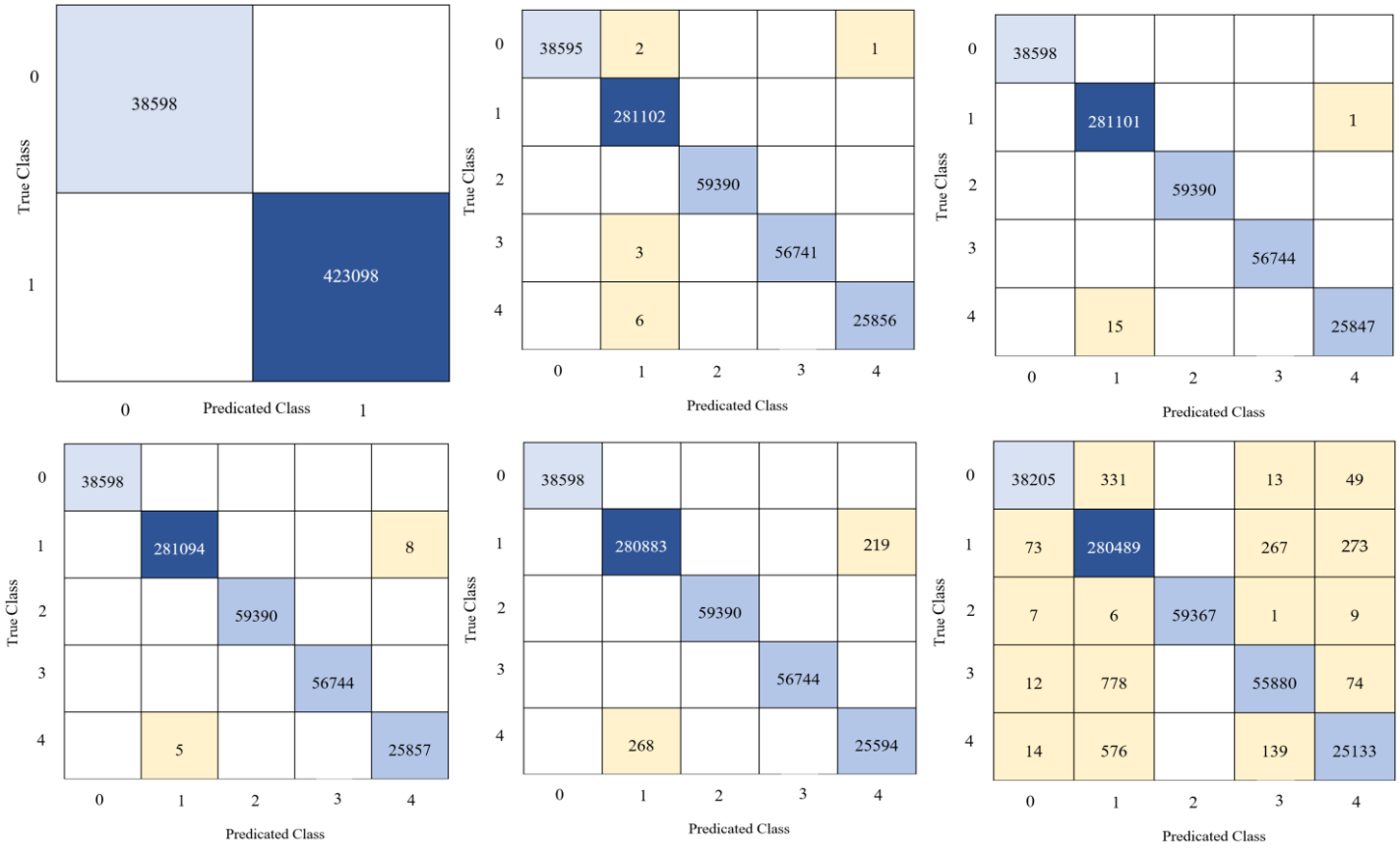


**Figure 13.** Confusion Matrix: (a) Detection model (For all models), (b) SNNs Classification model, (c) DT Classification model (d) BT Classification model, (e) SVM Classification model, (f) kNN Classification model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 October 2022 doi:10.20944/preprints202210.0431.v1

**Table 5.** Comparing our ML models' accuracy with existing ML models' accuracy.

| Research | Detection /Classification | ML Model | Accuracy |
|---|---|---|---|
| Sarwar et al. [16] | Detection | Random Forest | 98% |
| Sarwar et al.[16] | Classification | Random Forest | 83% |
| Song. et al. [17] | Classification | Auto-Encoders | 94.50% |
| Alkahtani et al. [18] | Classification | Convolutional Neural Networks + Long Short-Term Memory | 98.40% |
| Yang et al. [19] | Detection | LightGBM+ Optimized Adaptive Sliding Windowing | 99.9% |
| Al-Haija et al.[20] | Classification | Convolutional Neural Networks | 98.2% |
| Reddy et al.[21] | Classification | XGBoost | 99.7% |
| Proposed Method | Classification | Shallow Neural Networks | 100% |
| Proposed Method | Detection | Shallow Neural Networks | 100% |
| Proposed Method | Classification | Decision Trees | 99.9% |
| Proposed Method | Detection | Decision Trees | 100% |
| Proposed Method | Classification | Bagged Trees | 99.9% |
| Proposed Method | Detection | Bagged Trees | 100% |
| Proposed Method | Classification | Support Vector Machines | 99.80% |
| Proposed Method | Detection | Support Vector Machines | 100% |
| Proposed Method | Classification | K-Nearest Neighbor (KNN) | 99.40% |
| Proposed Method | Detection | K-Nearest Neighbor (KNN) | 100% |

## 5. Conclusions and Future work

This paper presents a new automated and intelligent intrusion detection system, modeled, implemented, and evaluated. The proposed predictive IDS utilizes machine learning techniques to detect and classify network activity in an IoT system. Particularly, five supervised learning models have been used, including shallow neural networks (SNNs), decision trees (DT), bagged tree (BT), support vector machine (SVM), and k-nearest neighbor (kNN). The developed models have been evaluated on a recent broad dataset known as the IoTID20. Additionally, the features' engineering approach was used with the dataset to increase the accuracy of the machine learning models. We used the confusion matrix metric to evaluate our models. As a result, our detection models recorded 100% for all machine learning models mentioned above. Furthermore, our classification models recorded 100% for the SNNs, DT, and BT, while KNN and SVM recorded 99.80% and 99.40%, respectively. Moreover, we will evaluate our predictive models with multiple IoT system datasets. In the future, we will seek to incorporate more datasets to develop a comparative study that compares the selected ML algorithms using several datasets. This will enrich the detection ability to detect more attack vectors in addition to those mentioned in this paper. Also, we believe that real-world deployment of the proposed IDS in different IoT/CPS networks (such as the internet of autonomous vehicles) is essential for more precise implementation representation and practical investigations. Furthermore, one can employ the deep neural networks or the log-linear neural networks [48] based intrusion detection system to provide deeper detection for the sub-categories of the stated attack vectors.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study

## References

[1] A.A. Smadi, B.T. Ajao, B.K. Johnson, H. Lei, Y. Chakhchoukh, *et al*, "Comprehensive survey on cyber-physical smart grid testbed architectures: requirements and challenges," *Electronics* 2021, vol.10, 1043.

[2] Q. Abu Al-Haija, M. Krichen, and W. Abu Elhaija, "Machine-learning-based darknet traffic detection system for IoT applications," *Electronics* 2022, vol.11, 556.

[3] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal *et al*, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721-82743, 2019 2019.

[4] A. Gharaibeh, "Smart cities: A survey on data management, security, and enabling technologies," *IEEE Communication Survey Tutorials*, vol. 19, no. 4, pp. 2456-2501, 2017.

[5] S. Ray, Y. Jin, and A. Raychowdhury, "The changing computing paradigm with the internet of things: A Tutorial Introduction," *IEEE Design and Test of Computers*, vol. 33, pp.76–96, 2016.

[6] Q. Abu Al-Haija, A.A. Smadi and M.F. Allehyani, "Meticulously intelligent identification system for smart grid network stability to optimize risk management," *Energies*, vol. 14, 6935, 2021.

[7] V.K. Quy, N.V. Hau, D.V. Anh, N.M. Quy, N.T. Ban *et al.*, "IoT-enabled smart agriculture: architecture, applications, and challenges," *Applied Sciences*, vol. 12, 3396, 2022.

[8] A. C. Jose and R. Malekian, "Improving smart home security: integrating logical sensing into smart home," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4269-4286, 2017.

[9] Q. Abu Al-Haija and J. Al-Saraireh, "Asymmetric identification model for human-robot contacts via supervised learning," *Symmetry*, vo.l. 14, 591, 2022.

[10] K. Albulayhi, A.A. Smadi, F.T. Sheldon and R.K. Abercrombie, "IoT intrusion detection taxonomy, reference architecture, and analyses," *Sensors*, vol. 21, 6432, 2021.

[11] R. Mahmoud, T. Yousuf, F. Aloul and I. Zualkernan, "Internet of things (IoT) security: current status, challenges, and prospective measures," in *Proc. ICITST*, London, UK, pp. 336–341, 2015.

[12] K. Albulayhi and F. T. Sheldon, "An adaptive deep-ensemble anomaly-based intrusion detection system for the internet of things," in *Proc.* AIIoT, Seattle, WA, USA, pp. 0187-0196, 2021.

[13] Q. Abu Al-Haija, "Top-down machine learning-based architecture for cyberattacks identification and classification in IoT communication networks," *Frontiers of Big Data*, vol. 4, 782902, 2022.

[14] Q. Abu Al-Haija and A. Ishtaiwi, "Machine learning based model to identify firewall decisions to improve cyber-defense," *International Journal on Advanced Science, Engineering and Information,* vol.11, no.4, pp. 1688 – 1695, 2021.

[15] R. Abdulhammed, M. Hassan, A. Ali, F.Miad and A. Abdelshakour, "Features dimensionality reduction approaches for machine learning-based network intrusion detection" *Electronics* 2019, 8, 322

[16] A. Sarwar, S. Hasan and W. U. Khan, "Design of an advance intrusion detection system for IoT networks," in *Proc. ICAI*, Islamabad, Pakistan, 2022.

[17] Y. Song, S. Hyun and Y.-G. Cheong, "Analysis of autoencoders for network intrusion detection," *Sensors*, vol. 21, 4294, 2021.

[18] H. Alkahtani and T. H. H. Aldhyani, "Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms," *Complexity*, vol. 2021, 5579851, 2021.

[19] L. Yang and A. Shami, "A lightweight concept drift detection and adaptation framework for IoT data streams," *IEEE Internet of Things Magazine*, vol. 04, no. 02, pp. 96 - 101, 2021.

[20] Q. Abu Al-Haija. and S. Zein-Sabatto, "An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks," *Electronics*, vol. 9, 2152, 2020.

[21] D.K. Reddy, H.S.Beheraa, J. Nayakb, B. Naikc, U. Ghoshd *et al*, "Exact greedy algorithm based split finding approach for intrusion detection in fog-enabled IoT environment." *Journal of Information Security and Applications* 60 (2021): 102866.

[22] K. Albulayhi, Q. Abu Al-Haija, S.A. Alsuhibany, A.A. Jillepalli, M. Ashrafuzzaman *et al*, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences*, vol. 12, 5015, 2022.

[23] A. Shahraki, M. Abbasi and Ø. Haugen, "Boosting algorithms for network intrusion detection: a comparative evaluation of real AdaBoost, Gentle AdaBoost and Modest AdaBoost," *Engineering Applications of Artificial Intelligence*, vol. 94, pp.10370-10380, 2021.

[24] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: network TON_IoT datasets," *Sustainable Cities and Society*, vol.72, 102994, 2021.

[25] V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif and E. A. Nasr, "Robust attack detection approach for IIoT using ensemble classifier," *Computers, Materials & Continua, vol. 66, no.3, pp.2457-2470, 2021*.

[26] Q. Abu Al-Haija and M. Al-Dala'ien, "ELBA-IoT: An ensemble learning model for botnet attack detection in iot networks," *Journal of Sensors and Actuator Networks (JSAN),* vol.11, no.12, pp.1-15. 2022.

[27] B.M.M. AlShahrani ,"Classification of cyber-attack using Adaboost regression classifier and securing the network." *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* vol. 12, no. 10, pp.1215-1223, 2021.

[28] X. Yang, L. David, X. Xia and J. Sun, "TLEL: A two-layer ensemble learning approach for just-in-time defect prediction," *Information and Software Technology (IST)*, vol. 87, pp. 206–220, 2017.

[29] Q. Abu Al-Haija and A. Al-Badawi, "Attack-Aware IoT network traffic routing leveraging ensemble learning," *Sensors*, vol, 22, 241, 2022.

[30] J. Ashraf, M. Keshk, N. Moustafa, M. Abdel-Basset, H. Khurshid *et al*, "IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities," *Sustainable Cities and Society*, vol.72, 103041, 2021.

[31] Q. Abu Al-Haija and A. Al Badawi and G.R. Bojja, "Boost-defence for resilient iot networks: a head-to-toe approach," *Expert Systems*, Early View e12934, 2022.

[32] B. Liao, Y. Ali, S. Nazir, L. He and H. U. Khan, "Security analysis of IoT devices by using mobile computing: a systematic literature review," *IEEE Access*, vol. 8, pp. 120331-120350, 2022.

[33] I. Ullah and Q. H. Mahmoud, "A scheme for generating a dataset for anomalous activity detection in IoT networks," in *Proc. Canadian Conference on Artificial Intelligence (CCAI)*, Ottawa, ON, Canada, p. 508–520, 2020.

[34] Q. Abu Al-Haija and A. A. Alsulami, "High-performance classification model to identify ransomware payments for heterogeneous bitcoin networks," *Electronics*, vol.10, 2113, 2021.

[35] S. Uddin, A. Khan, M. E. Hossain and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Medical Informatics and Decision Making*, vol. 19, 281, 2019.

[36] A. Derhab, A. Aldweesh, A. Z. Emam and F. A. Khan, "Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering," *Wireless Communications and Mobile Computing*, vol. 2020, p. 16, 2020.

[37] E. Shao, "Encoding IP address as a feature for network intrusion detection," *Purdue University Graduate School*, vol.1, 2019.

[38] H. Fang, P. Tang, and Hao Si, "Feature selections using minimal redundancy maximal relevance algorithm for human activity recognition in smart home environments," *Journal of Healthcare Engineering*, vol. 2020, 8876782, 2020.

[39] Z. Zhao, R. Anand, and M. Wang, "Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform," in *Proc. DSAA*, Washington, DC, USA, pp. 442-452, 2019.

[40] T. Ahmad and M. N. Aziz, "Data preprocessing and feature selection data preprocessing and feature selection," *ICIC Express Letters*, vol.13, no.2, pp. 93-101, 2019.

[41] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: a review of two decades of

research," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 97-116, 2017.

[42] B. T. Jijo and A. M. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 02, p. 20 – 28, 2021.

[43] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, *et al.*, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics - Proteomics*, vol. 15, no.1, pp. 41-51, 2018.

[44] A. R. Lubis, M. Lubis and Al-Khowarizmi, "Optimization of distance formula in k-nearest neighbor method," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no.1, pp. 326-338, 2020.

[45] F. Tchakounté and F. Hayata, "Supervised learning based detection of malware on android," *Mobile Security and Privacy: Advances, Challenges, and Future Research Directions*, vol.1, pp. 101-154, 2017.

[46] A. Y. Hussein, P. Falcarin and A. T. Sadiq, "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS," *Periodicals of Engineering and Natural Sciences*, vol. 9, pp. 579-591, 2021.

[47] Abdi, Abdulrahman, et al. "Multiclass classifiers for stock price prediction: a comparison study." *Journal of Harbin Institute of Technology* 54.3 (2022): 2022.

[48] H. Sun and R. Grishman, "*Lexicalized dependency paths based supervised learning for relation extraction*," *Computer Systems Science and Engineering*, vol. 43, no.3, pp. 861–870, 2022.