International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

http://dx.doi.org/10.12785/ijcds/110154

A Secure IoT Framework Based on Blockchain and Machine Learning

Rawan Shahin[1] and Khair Eddin Sabri[1]

[1]Computer Science department, The University of Jordan, Amman, Jordan

Received 10 Apr. 2021, Revised 17 Jul. 2021, Accepted 31 Dec. 2021, Published 31 Jan. 2022

Abstract: Internet of Things (IoT) network consists of many devices that communicate together and exchange data. IoT network has many applications especially in smart city and smart campus. IoT devices usually produce a huge amount of data that are stored in the cloud to be analyzed later. Data in general and IoT devices data in particular suffer from major security issues such as the availability and the integrity of data. Blockchain is a new technology that offers an interesting solution for the security of sensitive IoT data by protecting data against malicious tampering. However, the data stored in the blockchain cannot be altered, therefore, it should be validated before being stored in the blockchain especially that IoT devices are vulnerable to attacks. Machine learning algorithms are very useful to detect compromised IoT devices to ensure that only reliable data are stored in the blockchain.

The main aim of this paper is to propose a framework for storing reliable IoT data in a secure way that preserves the integrity and availability properties. First, to detect compromised IoT devices, several machine learning algorithms are evaluated. The IoTID20 dataset for anomalous activity detection in IoT network is used to build our machine learning model. IoTID20 dataset contains 80 network features and 625783 records. For our experiment, 4000 records are selected randomly from the dataset. Two algorithms are used to select features namely Logistic Regression and Pearson's correlation. A total of 15 features are selected in order to classify packets as (Normal, Anomaly). Several machine learning algorithms are trained: Logistic Regression, Random Forest, Decision Tree, K Nearest Neighbors, AdaBoost, and Naïve Bayes. These algorithms are compared based on the following measurements: accuracy, precision, recall, F1 score, and the time for classification. The results reveal that Random Forest and AdaBoost classifiers give very close results and are considered the best classifiers based on all performance metrics used in the paper except the time. Regarding the time required to detect malicious records, Decision Tree and Naïve Bayes are the best as the time required to predict 4000 records is 0.015 seconds.

A private blockchain is built to store normal data received from IoT devices after being filtered. The data is signed and then distributed to all nodes in the network for verification. To provide scalability, the consensus algorithm of the blockchain is based on the proof of authority algorithm. As a proof of concepts, the blockchain is developed using Python programming language and the time required to verify the signatures is computed.

Keywords: Blockchain; IoT; Machine Learning; Anomaly Detection

1. Introduction

Internet of Things (IoT) is a network of devices that exchange data. IoT can be used to automate processes which results in saving time and money. IoT devices, such as sensors, cameras, etc. usually collect data and then share them with a server for analysis and monitoring. The data stored in the server should be maintained in a way to preserve the integrity such that malicious attempt to modify the data is prevented. Furthermore, the data should be always available to other users and systems.

IoT network is used in many systems, and therefore, the number of IoT devices is increased accordingly. It is anticipated that the number of IoT devices would reach 25.44 billion in 2030 compared to 7.74 in 2019 [1]. The main problem with these devices is that security is not taken into consideration in most of them. Furthermore, when deployed, the username and passwords are not changed. Therefore, IoT devices such as cameras become the main target for attackers [2]. They try to compromise them and then use them as a botnet to steal information or be used for distributed denial-of-service (DDoS) attacks.

E-mail address: rshahin81@gmail.com, k.sabri@ju.edu.jo

This can be noticed in the number of attacks in the first half of 2019 on IoT devices estimated around 105 million [3]. As the number of attacks is increasing, more research is conducted to increase the security of IoT network and devices. For example, some authentication protocols are proposed by taking into consideration the limitation of resources in IoT devices [4] [5]. Others, propose different architectures design to increase the security such as [6] [7]. As the privacy of collected data is an important issue, algorithms are proposed to preserve the privacy [8].

IoT Network is usually connected with big data technology and algorithms. This is because IoT devices usually produce a large amount of data, and therefore, different algorithms are applied to get useful information and for automation [9]. Machine learning (ML) algorithms are used, as well, in security for building intrusion detection systems [10]. Technology plays an important role in handling and processing big data. For example, Apache Hadoop is an open source platform for processing and storing data. NoSQL (Non-relational Data Management System) is another example to handle and store big data [11].

Data collected from IoT devices are usually sent to servers to be stored and then analyzed. Usually, the data should be maintained in a way to preserve the integrity such that malicious attempt to modify the data is prevented. Furthermore, the data should be always available for other users and systems. Blockchain (BC) technology gives a solution for storing data in a secure way.

A BC technology is a distributed, decentralized and shared ledger. BC consists of a chain of connected blocks. When a new block is added, it contains the hash code of its previous one. The header of the block contains the hash code of the block, the hash of the previous block, timestamp and other information. The body of the block consists of transactions which are the main data of the BC. The values of these transactions are based on the application of the BC. The BC has received a great attention to apply this new technology in order to overcome the security concerns in many systems such as banking, voting, education, etc. [12].

While there are some advantages of using BC in IoT applications such as decentralization to eliminate a single point of failure, security enhancement, tractability, and immutability, using BC creates some challenges. In [13], the authors list some of these challenges when integrating BC and IoT such as storage, scalability, and vulnerability. One of the vulnerability issues is with the corrupted data before being stored in the BC. This issue comes from the immutability property of BC. Therefore, it becomes necessary to identify corrupted data before being sent and stored in the BC. Usually, corrupted data come from compromised devices. Therefore, ML techniques should be applied to detect such devices before storing data into BC and filter out their packets in order to store normal data only in the BC. However, as a huge amount of data should be stored in the BC and to overcome the scalability issue in BC, we adopt a proof of authority consensus algorithm. In this algorithm, no extra computation is required to solve a mathematical problem as in the proof of work algorithm.

In this paper, we propose a framework to store normal data in immutable storage that provides integrity and availability. The framework integrates BC and ML and then is applied to IoT network. The goal is to provide a secure way to store reliable IoT data from uncompromised devices. We design and implement a BC system to store data received from IoT devices. This system is integrated with ML to detect compromised devices and therefore, avoid writing corrupted data in the BC. We use the IoTID20 dataset [14] to build our detection system. We use Pearson correlation and Logistic Regression for feature selection and apply several classifiers to find the most suitable one. IoTID20 is a recent dataset that consists of normal and abnormal records in an IoT network. As performance is one of the critical issues in IoT network, we focus as well on selecting the most important features and measuring the time required to validate transactions and creating the block to measure the capacity of the framework.

The paper is organized as follows: Section 2 summarizes the most related papers to our work. Section 3 gives the architecture of the proposed framework. Section 4 focuses on the ML part and the detection of compromised devices. Section 5 presents the BC part of the framework. Finally, Section 6 concludes the paper and points to future work.

2. Literlature Review

The previous researches focused on investigating anomaly detection in IoT from many perspectives. Whereas others put emphasis on studying BC architecture of IoT systems. First, we present some works that combine both ML and BC. For example, Liang et. al., [15] proposed an intrusion detection system for IoT network. The system is based on BC, deep learning, and multi-agent system. Their system consists of four modules. The first module is for collecting data. The second module is for the management of the data. The third module is for the analysis and the final module is for the response. The NSL-KDD dataset is used to evaluate their system. Tanwar et. al., [16] developed a taxonomy covering the ML techniques required for BC. Also, they provided a case study to demonstrate the use of ML techniques in BC based system. Shen et. al., [17] presented a privacy preserving ML model. The authors proposed secureSVM which is a privacy preserving SVM classifier trained over encrypted IoT data. BC is used to provide a distributed way to share data among multiple providers. In [18], the authors presented a distributed ML for intrusion detection in IoT network. The IoT network is divided into autonomous systems which are monitored for intrusion detection. The support vector machine algorithm is trained and used for intrusion detection. BC is used for distributed sharing of attackers' information among the autonomous system.

Andročec and Vrček [19] presented a systematic review of ML in IoT security. In their paper, they state that support vector machine is the most used technique, and the main domain is the intrusion detection. The systematic review of the previous studies confirms that the use of ML for IoT security is a new and important research topic.

Access control in BC is one of the aspects that is investigated in the literature. For example, Liu et. al., [20] proposed a system based on the Hyperledger Fabric blockchain framework and attributed based access control for access control in IoT environment. Their system is called fabric-iot. Their aim is to specify and implement fine-grained access control policies in IoT network. Three contracts are implemented and the performance of the system is analyzed. Novo [21] proposed an architecture for access management. The architecture is based on BC and smart contracts. The aim of the paper is to address the scalability issue of managing access to a huge number of devices. Bao et. al., [22] presented three tiers of a blockchain-based architecture for IoT security. The first layer is for authentication and access control. The second is the BC layer which provides storage integrity. The last layer is an application layer. The performance of the architecture is evaluated. In [23], the authors proposed dynamic distributed security policies based on BC and ML. The BC is used to ensure the distributed aspect while the ML is used to provide dynamic security policies. In [24], the authors presented a survey on the integration of BC and ML. They show the benefits of ML on various aspects of BC such as big data processing, scalability and security.

As noted in the previous research, applying ML and/or BC to IoT network is a promising path. Our goal is different from that reported in the literature, which is applying ML to filter data before being stored in the BC. This is important as data cannot be changed after being stored in the BC. We also, focus on the performance of detecting abnormal behaviors and the time required to verify records before being mined and stored in blocks.

3. THE ARCHITURE OF THE FRAMEWORK

The main aim of the framework is to employ ML for detecting compromised IoT devices, and therefore, filter packets sent by these devices. Furthermore, the normal data coming from uncompromised devices is stored in the BC which supports decentralized architecture to IoT systems and enhances the security of preserving data.

The framework consists of four components and stages namely: IoT devices, Intruder Detection System (IDS), BC nodes, and BC network as shown in Figure 1. In the first stage, IoT devices collect data and then send them to the IDS for the second phase. The goal of IDS is to detect compromised devices and filter corrupted data. Then, in the third phase, the normal data is sent to the BC nodes. These nodes sign the data received from the IoT devices, after being filtered, and then sent to the BC network. Finally, in stage 4, the block that contains the data is created and sent to other nodes in the network to be added to the BC chain.

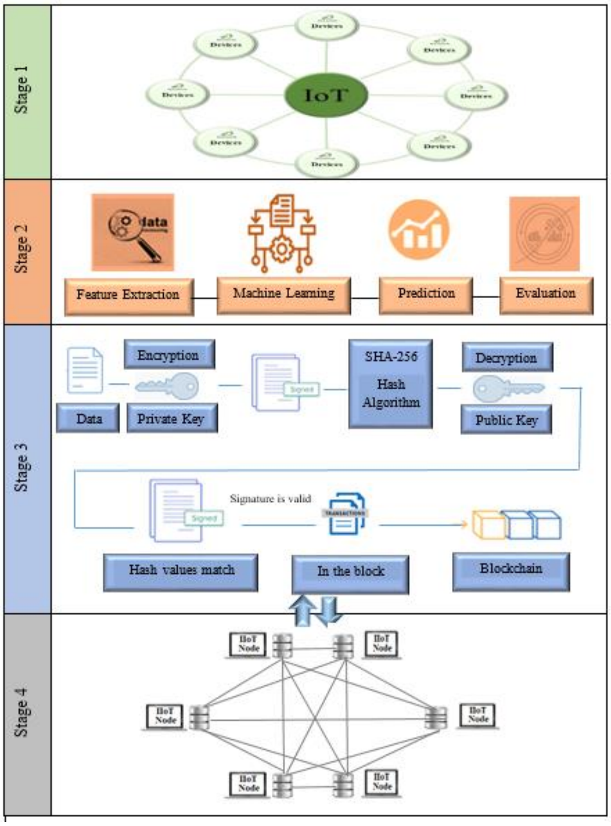


Figure 1. The stages within the framework

4. THE INTRUDER DETECTION SYSTEM

A. System description

We build the IDS based on different ML algorithms. We have analyzed and compared the following algorithms: Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN), AdaBoost, and Naïve Bayes (NB). These algorithms are selected as they are well known classifiers in IDS and are efficient in term of time [25] [26]. Other classifiers can be used as well.

As performance is one of the critical issues in IoT, we identity the most important features to be used for identifying compromised devices. Logistic Regression and Pearson's Correlation are used for feature selection. Features are the independent variables used to train the model, and predict the dependent variable. Below, we briefly describe each of these algorithms.

1. *Pearson Correlation*

Pearson Correlation is used for detecting the linear relationship between independent variables. Generally, the values of the correlation range between [–1, 1]. When the value is equal to 0 this means there is no relationship between the two variables. If the value of the correlation is negative i.e., less than 0 to –1 this means that the variables are negatively correlated. While if the value of correlation is positive i.e., more than 0 to 1, this indicates that the variables are positively correlated.

2. *Logistic Regression*

Logistic Regression (LR) is one of the popular classical statistical techniques. It can be defined as a non-linear method for modelling dichotomous dependent variables. The dichotomous variable is binary that has only two categories or classes. LR is used for classification and feature selection [27][28]. In order to select features, the independent variables (features) and the dependent variable (label) are entered into the model. Based on the value of P which reflects whether or not the particular feature contributes significantly to the occurrence of the outcome, features are selected. Usually, when the value of $P < 0.05$, the feature is considered statistically significant and it affects the probability of the target label outcome [29].

3. *Decision Tree*

Decision Tree (DT) is a ML technique. Generally, it consists of nodes, leaves and branches. The nodes represent features, the branches represent decision and the result is shown through the leaves. Developing decision trees take two stages, the first stage partitions the tree recursively. Once the tree has been built, a pruning process is performed to decrease the tree size. According to [30] [31], the main advantages of DT are dealing with continuous and discrete attributes, and handling missing attribute values. In addition, it decreases the tree size by pruning trees. Further, it saves the overall processing time since DT deals with variables without transformations.

4. *Random Forest*

Random Forest (RF) consists of multiple DT models that are combined together. These models are independent and are trained on different random subsets of training data.

5. *K-Nearest Neighbors*

KNN algorithm is one of the simplest techniques in ML. It classifies data based on similarity measurements.

6. *AdaBoost*

AdaBoost is a boosting ensemble model where models are added sequentially such that later models in the sequence correct the predictions made by earlier models. In this paper, a DT is used as the weak learners.

7. *Naïve Bayes*

The Naïve Bayes (NB) is a classifier based on the Bayes' Theorem with strong independence assumptions between features i.e., every pair of features is independent of each other.

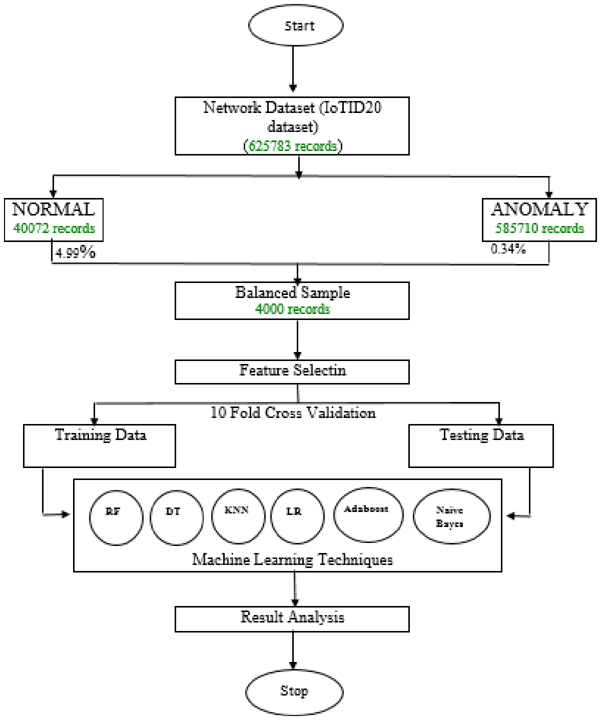


Figure 2. The Structure of the IDS

B. *IoTID20 dataset*

In our research, we use the IoTID20 dataset [32] to build our IDS system. IoTID20 is a new dataset that captures most of the recent attacks in IoT network. Data are collected from different IoT devices such as smartphones, tablets and laptops.

The dataset consists of 80 features. The records in the dataset are classified as normal and anomaly. The anomaly records are classified into four categories DoS, Mirai, MITM, and Scan. Each one of these records is further classified into subcategories as shown in Table 1 [14].

Table 1. IoTID20 Dataset Labels

Binary	Category	Subcategory
Normal	Normal	Normal
Anomaly	DoS	Syn Flooding
	Mirai	Brute Force, HTTP Flooding, UDP Flooding
	MITM	ARP Spoofing
	Scan	Host Port, OS

The IoTID20 dataset consists of 625783 records where 585710 records are anomaly and 40072 records are normal. The dataset reveals an imbalanced class based on the label feature. The imbalance class occurs when the majority of the records belong to one class. In this case, the classifier could be biased and gives inaccurate results. Figure 2 depicts the flowchart of ML processes that are implemented in the proposed framework.

C. Sample selection

Because the generated dataset is very large, it made handling the data slow. Therefore, 4000 records were selected from the original dataset. Figure 2 depicts the sample selection process. Since the original dataset is imbalanced, and in order to reach a balanced dataset, the whole dataset is divided into two sets based on the Label feature (Anomaly, Normal). Consistent with previous studies [33] [34], sampling of the dataset was used. The balanced sample of 4000 records consisted of 2000 normal records and 2000 anomaly records.

D. Preprocessing Data

The preprocessing of the sample data is essential to change the type and format of some features so that it can be used in the ML algorithm. For example, the label feature contains categorical values (Anomaly, Normal). Thus, the data feature was replaced with numerical values (0, 1). In addition, 4 features were removed as they are objects. These features are: Flow_ID, Src_IP, Dst_IP and Timestamp. Since the data sample was selected randomly, consequently, some features should be removed as it contains only one value. For example, Fwd_PSH_Flags feature for 4000 selected records has a single value equal to 0. Table 2 lists the 16 features removed because they contain a constant value.

Table 2. Constant Features

Features Name		
Fwd_PSH_Flags	CWE_Flag_Count	Bwd_Byts/b_Avg
Fwd_URG_Flags	ECE_Flag_Cnt	Bwd_Pkts/b_Avg
Bwd_URG_Flags	Fwd_Byts/b_Avg	Bwd_Blk_Rate_Avg
FIN_Flag_Cnt	Fwd_Pkts/b_Avg	Init_Fwd_Win_Byts
RST_Flag_Cnt	Fwd_Blk_Rate_Avg	Fwd_Seg_Size_Min
URG_Flag_Cnt		

E. Feature Selection

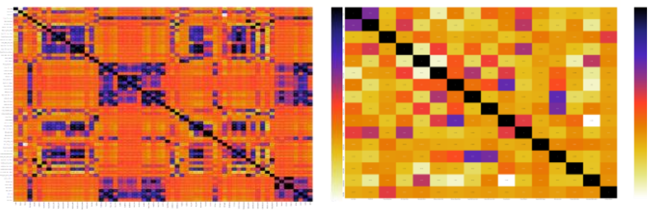
Selecting features is essential for any ML project as reducing the number of features reduces the computation cost, and therefore, improves the performance of the model [35].

1. *Filter method based on Pearson correlation*

The result of Pearson correlation is depicted in Figure 3. There is multicollinearity since the correlation coefficient among the independent variables exceeds 0.7 [36]. All the features were filtered out with Pearson's coefficient value in the range of [0, 0.70], only 16 features were selected. While the 47 highly correlated features were selected and then removed as shown in Table 3.

Table 3. The Removed Highly Correlated Features

Features Name		
ACK_Flag_Cnt	Dst_Port	Idle_Max
Active_Max	Flow_IAT_Max	Idle_Mean
Active_Min	Flow_IAT_Mean	Idle_Min
Active_Std	Flow_IAT_Min	Idle_Std
Bwd_Header_Len	Flow_IAT_Std	PSH_Flag_Cnt
Bwd_IAT_Max	Fwd_Act_Data_Pkts	Pkt_Len_Max
Bwd_IAT_Mean	Fwd_Header_Len	Pkt_Len_Mean
Bwd_IAT_Mean.1	Fwd_IAT_Max	Pkt_Len_Min
Bwd_IAT_Min	Fwd_IAT_Mean	Pkt_Len_Var
Bwd_IAT_Std	Fwd_IAT_Min	Pkt_Size_Avg
Bwd_IAT_Tot	Fwd_IAT_Tot	SYN_Flag_Cnt
Bwd_Pkt_Len_Max	Fwd_Pkt_Len_Max	Subflow_Bwd_Byts
Bwd_Pkt_Len_Mean	Fwd_Pkt_Len_Mean	Subflow_Bwd_Pkts
Bwd_Pkt_Len_Min	Fwd_Pkt_Len_Min	Subflow_Fwd_Byts
Bwd_Pkts/s	Fwd_Pkts/s	Subflow_Fwd_Pkts
Bwd_Seg_Size_Avg	Fwd_Seg_Size_Avg	

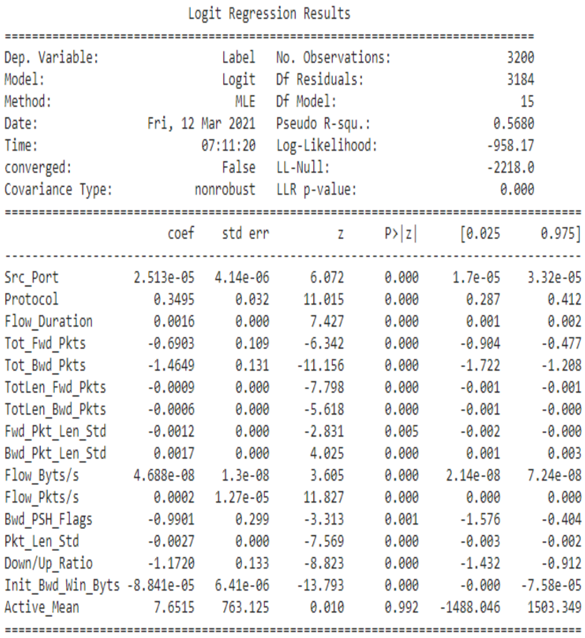


(a) (b)

Figure 3. Pearson Correlation Result (a) before filter (b) after Filter

2. *Features Selection with Logistic Regression Model*

After removing the highly correlated features depicted in Table 3, the remaining features were tested using Logistic Regression in order to determine the significant ones. The output of Logistic Regression is shown in Figure 4, which depicts all of the following features that are significant at 0.05.



```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Label   No. Observations:                 3200
Model:                          Logit   Df Residuals:                     3184
Method:                           MLE   Df Model:                           15
Date:                Fri, 12 Mar 2021   Pseudo R-squ.:                  0.5680
Time:                        07:11:20   Log-Likelihood:                -958.17
converged:                      False   LL-Null:                       -2218.0
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
Src_Port           2.513e-05   4.14e-06      6.072      0.000     1.7e-05    3.32e-05
Protocol              0.3495      0.032     11.015      0.000       0.287       0.412
Flow_Duration         0.0016      0.000      7.427      0.000       0.001       0.002
Tot_Fwd_Pkts         -0.6903      0.109     -6.342      0.000      -0.904      -0.477
Tot_Bwd_Pkts         -1.4649      0.131    -11.156      0.000      -1.722      -1.208
TotLen_Fwd_Pkts      -0.0009      0.000     -7.798      0.000      -0.001      -0.001
TotLen_Bwd_Pkts      -0.0006      0.000     -5.618      0.000      -0.001      -0.000
Fwd_Pkt_Len_Std      -0.0012      0.000     -2.831      0.005      -0.002      -0.000
Bwd_Pkt_Len_Std       0.0017      0.000      4.025      0.000       0.001       0.003
Flow_Byts/s        4.688e-08     1.3e-08      3.605      0.000    2.14e-08    7.24e-08
Flow_Pkts/s           0.0002   1.27e-05     11.827      0.000       0.000       0.000
Bwd_PSH_Flags        -0.9901      0.299     -3.313      0.001      -1.576      -0.404
Pkt_Len_Std          -0.0027      0.000     -7.569      0.000      -0.003      -0.002
Down/Up_Ratio        -1.1720      0.133     -8.823      0.000      -1.432      -0.912
Init_Bwd_Win_Byts -8.841e-05   6.41e-06    -13.793      0.000      -0.000   -7.58e-05
Active_Mean           7.6515    763.125      0.010      0.992   -1488.046    1503.349
=====================================================================================
```

Figure 4. Logistic Regression Results

Based on the Logistic Regression results the following features significantly affect the dependent variable and are selected for building the ML model.

Table 4. The selected features

Features Name		
Src_Port	Protocol	Flow Duration
Tot_Fwd_Pkts	Tot_Bwd_Pkts	TotLen_Fwd_Pkts
TotLen_Bwd_Pkts	Fwd_Pkt_Len_Std	Bwd_Pkt_Len_Std
Flow_Byts/s	Flow_Pkts/s	Bwd_PSH_Flags
Pkt_Len_Std	Down/Up_Ratio	Init_Bwd_Win_Byts

The Active_Mean feature does not significantly affect the dependent variable (label) (*P*=0.992), so it was removed. Therefore, these features can be used to train the ML algorithms that are used in this paper for classification.

F. Performance Measures for Machine Learning

In this paper, we used four measurements to compare between the different algorithms used in the paper. The first measurement is the accuracy which is the ratio of the total number of samples predicted correctly to the total number of samples as shown in equation (1). The second measurement is the precision which is the ratio of true positive to the sum of true positive and false positive as shown in equation (2). The third measurement is the recall which is the ratio of true positive to the sum of true positive and false negative as shown in equation (3). Finally, the fourth measurement is the F1 score which considers the calculation results of the precision and recall measurements as shown in equation (4). In all these measurements, the larger the value is, the better the model is. Following are the equations used to compute each one of these measures where TP represents true positive, TN for true negative, FP for false positive, and FN for false negative.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-Score} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4}$$

G. Results

The IoTID20 dataset is analyzed and evaluated depending on different ML algorithms for classification. The 10-fold cross validation technique is used to evaluate the ML models, where the dataset is partitioned into ten groups such that nine of them are used for training and one is used for testing. This process is repeated ten times and the average is computed. Then, the four measurements mentioned previously are used to measure the performance of each model.

The paper builds different models based on the dataset and the six classification algorithms. The aim of the model is to classify records as normal or anomaly. Then, the accuracy, precision, recall and F1 score are computed for each model as given in the following subsections.

1. *Accuracy*

From Table 5 and Figure 5, the average value of accuracy for LR equals to 85.4%, DT equals to 96%, RF equals to 96.2% KNN equals to 82.4%, Adaboost equals to 96.3% and NB equals to 69.2. These results indicate that the DT, RF, and AdaBoost are the best classifiers for intrusion detection based on accuracy results.

Table 5. The Accuracy of the Machine Learning Algorithms

Round	LR	DT	RF	KNN	Adaboost	NB
1	84.0	94.8	95.8	83.3	95.5	71.3
2	85.5	93.5	95.0	84.0	95.3	65.0
3	87.3	97.8	97.5	84.0	97.5	71.5
4	84.5	96.8	96.8	84.0	96.8	69.3
5	86.3	95.5	94.8	82.5	95.3	70.8
6	89.3	96.0	96.3	81.3	96.5	67.8
7	86.5	96.5	97.0	82.8	97.0	68.3
8	82.5	96.3	96.0	79.3	96.8	67.3
9	86.0	95.8	95.8	80.5	95.5	71.0
10	82.5	97.5	97.0	82.5	97.3	69.5
Average	85.4	96.0	96.2	82.4	96.3	69.2

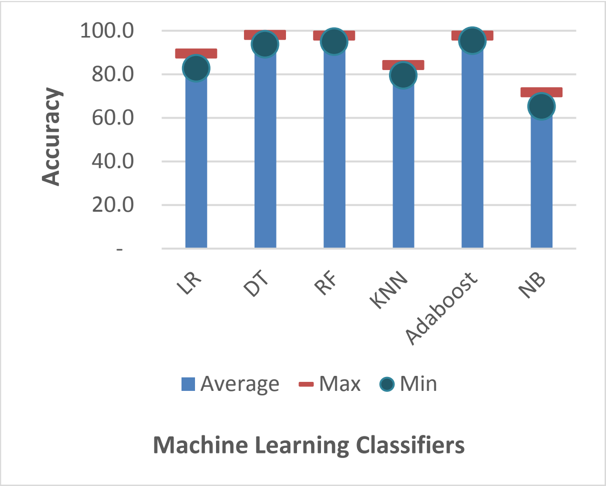


Figure 5. The Accuracy of Machine Learning Algorithms

2. *Precision*

From Table 6 and Figure 6, the average value of precision for LR equals to 85.8%, DT equals to 97.6%, RF equals to 96.2% KNN equals to 84.4%, Adaboost equals to 97.9% and NB equals to 94.9%. These results indicate that the AdaBoost is the best classifier in terms of precision.

Table 6. The Precision of the Machine Learning Algorithms

Rounds	LR	DT	RF	KNN	Adaboost	NB
1	82.4	96.2	94.8	83.5	96.7	96.3
2	89.9	98.0	98.5	88.4	100.0	96.4
3	86.6	98.4	95.5	86.0	98.9	97.6
4	83.4	97.0	96.5	83.6	97.0	94.2
5	85.2	96.3	93.4	84.7	95.3	94.2
6	90.2	98.0	97.5	84.5	98.0	94.4
7	85.6	97.7	96.7	82.8	97.8	90.0
8	83.6	97.2	95.1	83.3	98.6	92.2
9	85.1	97.8	96.3	81.1	97.3	97.5
10	85.5	99.5	97.6	86.6	99.5	96.7
Average	85.8	97.6	96.2	84.4	97.9	94.9

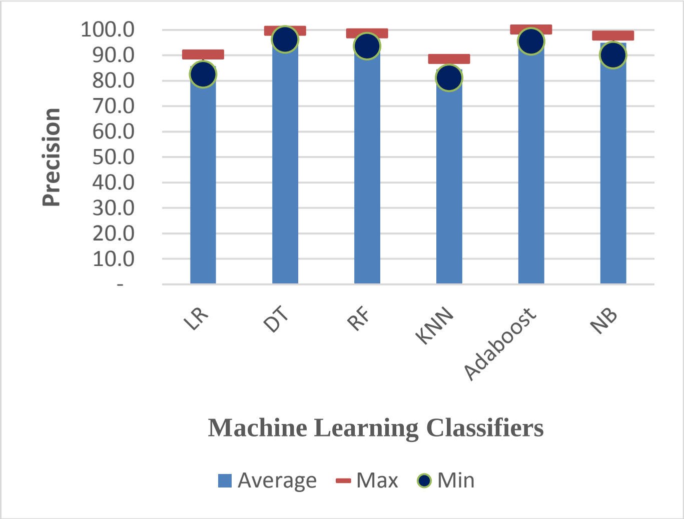


Figure 6. The Precision of Machine Learning Algorithms

3. *Recall*

From Table 7 and Figure 7, the average value of precision for LR equals to 85%, DT equals to 94.4%, RF equals to 96.2% KNN equals to 79.4%, Adaboost equals to 95% and NB equals to 46.4%. These results indicate that the RF is the best classifiers in terms of recall.

Table 7. The Recall of the Machine Learning Algorithms

Round	LR	DT	RF	KNN	Adaboost	NB
1	84.1	92.6	96.3	80.4	93.7	40.7
2	82.5	89.9	92.2	81.1	91.2	96.4
3	87.0	96.9	99.5	79.8	95.9	42.0
4	85.9	96.5	97.0	84.4	96.5	40.7

5	86.5	94.3	95.9	77.7	94.8	42.0
6	88.9	94.2	95.2	78.4	95.2	40.4
7	84.7	94.5	96.7	78.7	95.6	34.4
8	84.4	95.9	97.7	77.5	98.6	43.6
9	86.0	93.3	94.8	77.7	93.3	40.9
10	79.7	95.7	96.6	78.3	95.2	42.5
Average	85.0	94.4	96.2	79.4	95.0	46.4

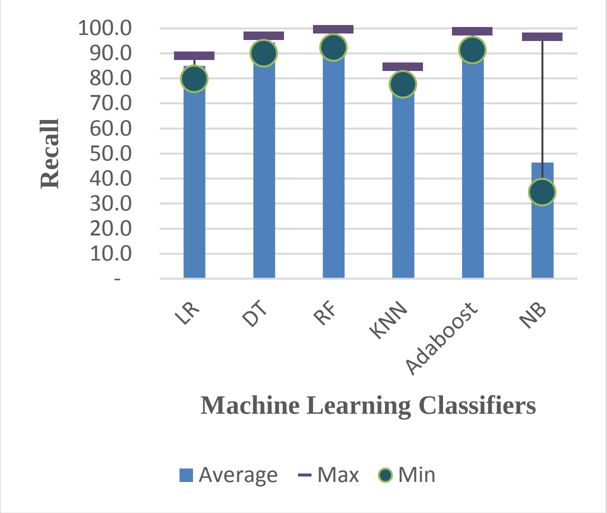


Figure 7. The Recall of the Machine Learning Algorithms

4. *F1 – score*

From Table 8 and Figure 8, the average value of the F1-score shows that Adaboost gets the highest F1 score (96.3%) while NB gets the lowest score (56.6%). The value of LR is equal to 85.3%, the value of DT is equal to 96%, the value of RF is equal to 96.2%, and the value of KNN is equal to 81.8%.

Table 8. The F1-score for Machine Learning Algorithms

Round	**LR**	**DT**	**RF**	**KNN**	**Adaboost**	**NB**
1	83.2	94.3	95.5	81.9	95.2	57.2
2	86.1	93.7	95.2	84.6	95.4	53.3
3	86.8	97.7	97.5	82.8	97.4	58.7
4	84.7	96.7	96.7	84.0	96.7	56.8
5	85.9	95.3	94.6	81.1	95.1	58.1
6	89.6	96.1	96.4	81.3	96.6	56.6
7	85.2	96.1	96.7	80.7	96.7	49.8
8	84.0	96.5	96.4	80.3	97.0	59.2
9	85.6	95.5	95.6	79.4	95.2	57.7
10	82.5	97.5	97.1	82.2	97.3	59.1
Average	85.3	96.0	96.2	81.8	96.3	56.6

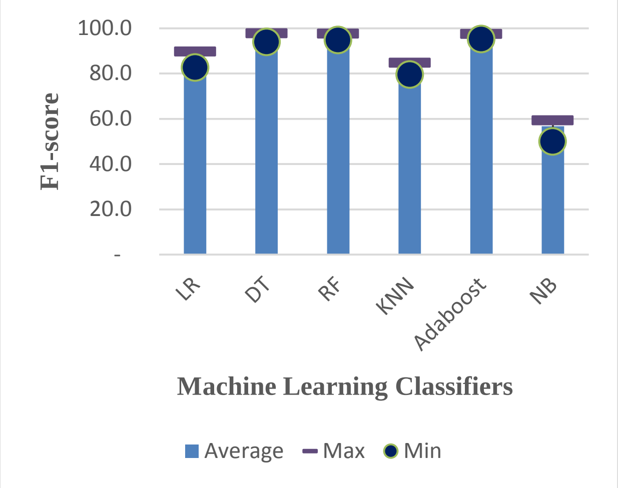
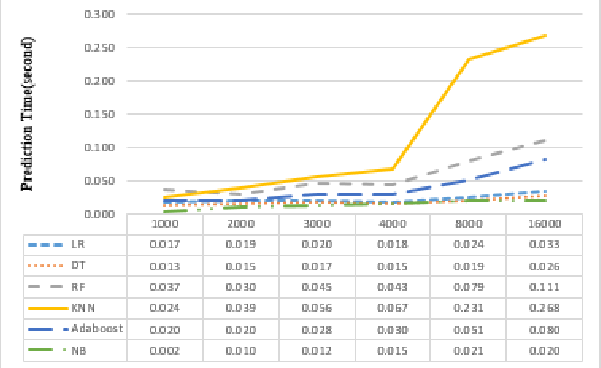


Figure 8. F1-score for Machine Learning Algorithms

5. *Performance*

Figure 9 depicted the relationship between the sample size and prediction time for the used classifier in the proposed model. The results show that the prediction time increases as the sample size increases for all classifiers. In addition, the prediction time for all six classifiers is approximately equal when the sample size is small. However, by increasing the sample size, the DT and NB classifiers are the best.



	1000	2000	3000	4000	8000	16000
LR	0.017	0.019	0.020	0.018	0.024	0.033
DT	0.013	0.015	0.017	0.015	0.019	0.026
RF	0.037	0.030	0.045	0.043	0.079	0.111
KNN	0.024	0.039	0.056	0.067	0.231	0.268
Adaboost	0.020	0.020	0.028	0.030	0.051	0.080
NB	0.002	0.010	0.012	0.015	0.021	0.020

Figure 9. Performance Metrics of Time Using Machine Learning Algorithms

5. BLOCKCHAIN

The other component of the proposed framework is related to preserving data in an immutable and secure way. The BC nodes play an important role. It is their responsibility to send signed transactions, verify transactions sent by other nodes, and validate blocks through the mining process. The private BC is assumed in our framework so that only authorized participants are able to access the BC. The processes of BC are summarized by the following steps:

1. BC nodes receive the packet after being filtered. Note that IDS and BC node can be installed on the same device.
2. The BC node signs transactions and then sends them to other nodes in the network.
3. In BC, all nodes are synchronized together so that they all receive the signed transactions and verify them.
4. Miner nodes build a BC block from the signed transaction, and then send it to the network.
5. Each BC node verifies the block and then adds it to the chain.

A. *Architecture*

BC is a decentralized system that is continually updated and kept synchronized. In addition, it is distributed across peer-to-peer network where the data is distributed over all nodes. In this section, we discuss the main components of the BC.

1. *Transactions*

The BC nodes process the collected normal filtered IoT data from the IDS. Then, they send the data as a packet called a transaction. Each transaction contains specific data such as IoT device Id, IP Address, Data, Timestamp etc. Each BC node has a private key used with a hash function to sign the transactions.

2. *Transactions Pool*

Every BC node verifies the transactions received from other nodes. The node uses both a hash SHA-256 function and the public key of the sender to verify the digital signature of transactions, to check if the transaction has been tampered or modified. The verified transactions are moved to the transaction pool where transactions are still not mined.

3. *Mining Process*

In the mining process, a block is created and then transactions are added to the block after being verified. There are different consensus algorithms in the literature. As the proposed BC is private and as the performance and scalability are important aspects to be taken into consideration in the framework, a Proof of Authority (PoA) consensus algorithm is used in the process. PoA is an efficient and practical algorithm that is suitable in the private BC. It does not require to solve a complex mathematical problem, instead it is based on a set of notes called authorities to validate transactions and build blocks. In our model, as we consider a limited number of nodes, all the BC nodes belong to the set of authorities. However, in other applications, a subset of the nodes can be selected based on selection criteria. The other nodes in the network would validate the block and in case it is valid, it would accept the block and add it to the chain.

4. *Blocks*

Each block consists of two parts: the header and the body. Block header consists of various fields. One of which is the hash value of the previous block. The second field is the hash value used as the identifier of the current block. The other fields are the timestamp, nonce and the Merkle root. Blocks are linked back through the value of the previous hash. Therefore, any modification in one block must be reflected to the hash value of other blocks since the subsequent block contains the hash value of the previous block. The structure of any block in BC is shown in Figure 10.

5. *Blockchain*

BC is a sequence of chain blocks. Blocks are connected by storing the hash code of the previous block. The design of BC prevents tampering. In the BC network, each participant node has a copy of each block and keeps other participants' nodes synchronized. This means in BC network, once data is stored in the block, it becomes difficult to be tampered because all of the distributed copies must be attacked simultaneously.

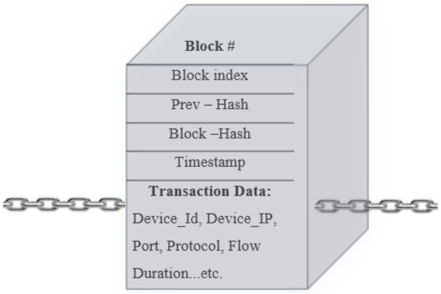


Figure 10. An Example of a block

B. *Implementation*

In this section, we briefly give the implementation of the main components of the BC. For illustration, we assume that the number of nodes is 6, this number can be generalized to n nodes.

1. *Digital Signature*

SHA256 hash function and Elliptic Curve Cryptography (ECC) are used to provide integrity and digital signature to the data stored in BC. Each node has a private key and a public key.

```
-----PRIVATE KEY VALUE-----
0x63bd3b01c5ce749d87f5f7481232a93540acdb0f7b5c014ecd
9cd32b041d6f33
```

```
-----PUBLIC KEY VALUE-----
04017655e42a892cc71bccedcb1cd421d03530e1d7edb52cef14
3c5562c4c6f0129fa5a37738013e64a1ff0e6cb7068815a13000
eb162cb7a0214dfcf3c8fa101c
```

As mentioned earlier, transactions sent by a BC node are digital signed using its private key which can be verified using the public key. Elliptic curve cryptography was used to generate digital signatures. The following screen shows the transaction was digitally signed and the value of the digital signature.

```
The Transaction is signed successfully...
Signature:
0x304502205c90f8e4ac34695528b72e9a910d41dfb9ee82cfa9
0a2642eb5c1b7cc1ebcd0e022100b69e19219ca6cd54329c5f9d
48c7ee60e93c598339f1c8c1650269bb21e1f751
```

2. Hash Function

The hash SHA-256 algorithm was used in the proposed model. Hashing is used to generate a signature of a text. The output of SHA-256 is 256-bit. Hash value is an essential part of building blocks in BC. Each block is verified separately through its hash value. The following diagram shows the blocks within the BC linked back by referring to the previous block hash value.

```
The genesis block has been created.
Timestamp: 2021-03-12 19:35:40.477866
Pre-
Hash:00000000000000000000000000000000000000000000000000
0000000
Hash:
4877028dd1f5afbdea14441f1a54d85a8bee003253a446328329c8de38
30c3b3
==========================================================
Block #1 created.
Timestamp: 2021-03-12 19:35:40.479876
Pre-Hash:
4877028dd1f5afbdea14441f1a54d85a8bee003253a446328329c8de38
30c3b3
Hash:
f2e915db8f755d2bec57a12e21d69c8055090bcc1ff0c4c052f369cdd4a7
dcde
==========================================================
Block #2 created.
Timestamp: 2021-03-12 19:35:40.481885
Pre-Hash:
f2e915db8f755d2bec57a12e21d69c8055090bcc1ff0c4c052f369cdd4a7
dcde
Hash:
74de5cc56014a9b4eda3faed9dbfca2f872d5d9df2d87a7437726e23137
87412
==========================================================
Block #3 created.
Timestamp: 2021-03-12 19:35:40.482886
Pre-Hash:
74de5cc56014a9b4eda3faed9dbfca2f872d5d9df2d87a7437726e23137
87412
Hash:
00a9d4275e91263ca9ef9ccfa033736aa3732f09ee6ddb606fc3f6996b0
71160
==========================================================
Block #4 created.
Timestamp: 2021-03-12 19:35:40.484877
Pre-Hash:
00a9d4275e91263ca9ef9ccfa033736aa3732f09ee6ddb606fc3f6996b0
71160
Hash:
2433859501d9306d1a0faedfbd31792b46278cbc81a5c368e5bcf856d5
1e1f68
==========================================================
Block #5 created.
Timestamp: 2021-03-12 19:35:40.485876
Pre-Hash:
2433859501d9306d1a0faedfbd31792b46278cbc81a5c368e5bcf856d5
1e1f68
Hash:
28eb2db1e1a9251c5aaa3fa41ff665979dc380eadea69ec3357c81b712b
fb68f
==========================================================
Block #6 created.
Timestamp: 2021-03-12 19:35:40.487871
Pre-Hash:
28eb2db1e1a9251c5aaa3fa41ff665979dc380eadea69ec3357c81b712b
fb68f
Hash:
3b939a8bf5da1cbbaf2b3496fec3706332ee375d6a7eda17c4d06a7fc7a
e5e89
==========================================================
```

C. Properties of the Proposed Framework

The proposed framework has many properties inherited by the use of BC. These properties are summarized in this section based on the following assumptions:

A. Each BC nodes has a private key that is not compromised and a public key shared with other nodes.

B. The used hash function is secure against collusion attacks such that it is computationally infeasible to find x and y such that $x \neq y$ and $H(x) = H(y)$.

C. The communication between the IoT devices and the BC node is secured. This can be achieved through different approaches, one of them is having a secured channel where encryption is used with pre-shared keys. However, this is out of the scope of the paper, and therefore, we assume the communication is secure.

D. The majority of BC nodes are trusted. As our BC is a private network, therefore, the ability to access the BC network should be controlled. Therefore, this assumption is valid.

E. There is an access control mechanism implemented in the BC network in order to restrict access to the network to the authorized nodes only. Several access control models are proposed in the literature which can be used to guarantee unauthorized access to the network is prohibited.

1. *Integrity*

The integrity is the trustfulness of data; such that data is not altered by unauthorized user. The data can be changed either by transition or when stored. During transition, there are three communications which are:

IoT devices to Blockchain node: As discussed previously, it is one of our assumption which can be achieved by using secure channel through pre-shared key and using a secure hash function.

Blockchain node to the network: The data after being processed by the BC node and signed, it will be distributed through the network. The nodes should verify the signature and if it is valid it will be spread in the network, otherwise it will be dropped. Therefore, invalid transactions are not accepted. However, if the BC node is compromised, there is a possibility of inserting fake data to the BC.

Miner to the network: After the block is mined, it will be spread to the network to be added to the BC. All the nodes should verify the block and therefore, make sure it is valid.

After data being stored in BC it cannot be changed as BC is distributed and immutable to changes.

2. *Availability*

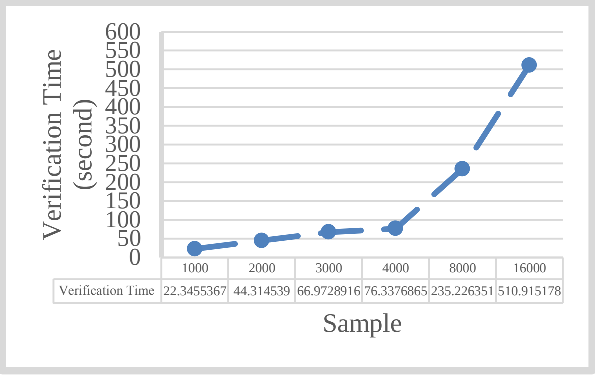
In the proposed system, data are stored by BC nodes. So the availability of data is achieved by making data available whenever it is required from any node.

3. *Performance*

Two factors affect the performance of our framework, which we analyze in the section

- Time required to filter the records
- Time required for digital signature and validation of blocks and transactions.

The time required for classification is analyzed and is presented in Figure 9. In the proposed framework, and to increase the performance, Elliptic Curve digital signature is used to sign and verify transactions. It is known that ECC has better performance over other techniques such as RSA. The main time is required to validate the transaction is to check the signature. The verification time was computed for different number of transactions. Figure 11 shows that the verification time increases as a result of increasing the sample size.



	1000	2000	3000	4000	8000	16000
Verification Time	22.3455367	44.314539	66.9728916	76.3376865	235.226351	510.915178

Figure 11. Relation Between Sample Size and Verification Time

6. CONCLUTION AND FUTURE WORK

IoT devices are playing a crucial role in our modern life. Along with the wide use of IoT devices in various sectors, the IoT system suffers from many malicious attacks. This paper is designed to introduce state of the art solution related to security issues and decentralization.

In order to introduce a solution to the problem at hand, the proposed framework was designed to enhance security for detecting malicious attacks. The framework consists of four components namely: IoT devices, IDS, BC nodes and BC Network.

Four main aspects were handled in this paper to achieve secure, fast and accurate prediction of anomaly detection model. The first aspect of this paper highlights the importance of selecting features depending on appropriate ML models. The experiment was conducted using two different ML algorithms namely Pearson correlation and Logistic Regression on IoTID20 dataset which consists of 80 features. Based on the experiment results, 15 features were selected for anomaly detection. This process can minimize computationally intensive models to reach fast-time prediction models.

The second aspect of this paper is to build a classifier based on ML algorithms and specifically DT, RF, LR, KNN, AdaBoot, and NB. Then, comparison between these algorithms is made based on many performance metrics including accuracy, precision, recall and F1 score. The experiment shows that RF and AdaBoost classifiers are the best among others based on performance metrics used in the paper where the accuracy of AdaBoost equals to 96.3%, the precision equals to 97.9%, the recall equals to 95% and the F1 score equals to 96.3%. Whereas, the accuracy of RF equals to 96.2%, the precision equals to 96.2%, the recall equals to 96.2% and the F1 score equals to 96.2%

The third aspect is to build secure storage by using BC technology which mainly consists of two facets. The first facet is to employ the digital signature of the trusted BC nodes using the private key to verify tampering transactions. The second facet is to hash the entry data to increase security.

The fourth aspect is analyzing the performance for intruder detection and transactions verification. The result shows a positive relationship between the sample size and prediction time. Results from the experiments show that DT and NB classifiers are the best ones based on time prediction.

As the developed IDS is not 100% accurate, there is a possibility of rejecting normal data and accepting corrupted data. However, all rejected data can be further analyzed for assuring that it is corrupted and in case it is normal, it can be added to the BC. Handling the other case is more challenging as writing corrupted data to BC cannot be removed or updated. One way to deal with this challenge is to enhance the IDS to minimize the number of false negative cases i.e., minimize the number of unrecognized anomaly records.

As future work, we are planning to enhance IDS by considering other algorithms and datasets. Furthermore, investigating other BC platforms such as Ethereum and Microsoft Azure services based on many performance measures as efficiency. Also, investigating other structures such as storing the data off-chain while storing summary of the data on the BC.

REFERENCES

[1] Statista Research Department. "IoT: Number of connected devices worldwide 2019–2030". Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ [Accessed on 03-June-2021].

[2] N. R. Techniques, H. D. D. Expose, A. Target, and M. Lucrative, “McAfee Labs Threats Report: December 2018,” *Comput. Fraud Secur.*, vol. 2019, no. 1, p. 4, 2019, doi: 10.1016/s1361-3723(19)30004-1.

[3] “IoT under fire: Kaspersky detects more than 100 million attacks on smart devices in H1 2019 | Kaspersky.” [Online]. Available: https://www.kaspersky.com/about/press-releases/2019_iot-under-fire-kaspersky-detects-more-than-100-million-attacks-on-smart-devices-in-h1-2019. [Accessed on 03-June-2021].

[4] M. Saadeh, A. Sleit, K. E. Sabri, and W. Almobaideen, “Object Authentication in the Context of the Internet of Things: A Survey,” *Journal of Cyber Secuity and Mobility*, vol. 9, pp. 385–448, 2020, doi: 10.13052/jcsm2245-1439.932.

[5] M. Saadeh, A. Sleit, K. E. Sabri, and W. Almobaideen, “Hierarchical architecture and protocol for mobile object authentication in the context of IoT smart cities,” *Journal of. Network and Computer Application*, vol. 121, pp. 1–19, 2018, doi: 10.1016/j.jnca.2018.07.009.

[6] H. Saadeh, W. Almobaideen, K. E. Sabri, and M. Saadeh, “Hybrid SDN-ICN architecture design for the internet of things,” *2019 6th Internation Conference on Software Defined. Systems SDS 2019*, pp. 96–101, 2019, doi: 10.1109/SDS.2019.8768582.

[7] H. Saadeh, W. Almobaideen, and K. E. Sabri, “Internet of Things: A review to support IoT architecture’s design,” *Proc. 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes and Systems IT-DREPS 2017*, vol. 2018-Janua, pp. 1–7, 2018, doi: 10.1109/IT-DREPS.2017.8277803.

[8] H. K. Saadeh, W. Almobaideen, and K. E. Sabri, “PPUSTMAN: Privacy-Aware PUblish/Subscribe IoT MVC Architecture Using Information Centric Networking,” *Modern Appied. Science.*, vol. 12, no. 5, p. 128, 2018, doi: 10.5539/mas.v12n5p128.

[9] I. Machorro-Cano, G. Alor-Hernández, M. A. Paredes-Valverde, L. Rodríguez-Mazahua, J. L. Sánchez-Cervantes, and J. O. Olmedo-Aguirre, “HEMS-IoT: A big data and machine learning-based smart home system for energy saving,” *Energies*, vol. 13, no. 5, 2020, doi: 10.3390/en13051097.

[10] H. Alazzam, A. Sharieh, and K. E. Sabri, “A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer,” *Expert Systems with Applications*, vol. 148, 2020, doi: 10.1016/j.eswa.2020.113249.

[11] B. Balusamy, N. Abirami R, S. Kadry, and A. H. Gandomi, "Big Data Concepts Technology and Architecture", Wiley 1st edition, 2011

[12] A. Shorman, K. E. Sabri, M. Abushariah and M. Qaimari, "Blockchain for banking systems: Opportunities and challenges," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 23, p. 3703–3717, 2020

[13] N Waheed, X. He, M. Ikram, M. Usman, S. Hashmi, and M. Usman, "Security and Privacy in IoT Using Machine Learning and Blockchain: Threats and Countermeasures," *ACM Computing Surveys*, vol. 53, issue 6, pp 1–37, 2021.

[14] I. Ullah and Q. H. Mahmoud, “A Technique for Generating a Botnet Dataset for Anomalous Activity Detection in IoT Networks,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 2020-Octob, no. May, pp. 134–140, 2020, doi: 10.1109/SMC42975.2020.9283220.

[15] C. Liang *et al.*, “Intrusion detection system for the internet of things based on blockchain and multi-agent systems,” *Electron.*, vol. 9, no. 7, pp. 1–27, 2020, doi: 10.3390/electronics9071120.

[16] S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, P. K. Singh, and W. C. Hong, “Machine Learning Adoption in Blockchain-Based Smart Applications: The Challenges, and a Way Forward,” *IEEE Access*, vol. 8, pp. 474–448, 2020, doi: 10.1109/ACCESS.2019.2961372.

[17] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, “Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7702–7712, 2019, doi: 10.1109/JIOT.2019.2901840.

[18] M. A. Cheema, and H. K. Qureshi, "Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things", *16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2020.

[19] D. Andročec and N. Vrček, “Machine learning for the internet of things security: A systematic review,” *ICSOFT 2018 - Proc. 13th International Conference of Software Technologies*, pp. 563–570, 2019, doi: 10.5220/0006841205630570.

[20] H. Liu, D. Han, and D. Li, “Fabric-iot: A Blockchain-Based Access Control System in IoT,” *IEEE Access*, vol. 8, pp. 18207–18218, 2020, doi: 10.1109/ACCESS.2020.2968492.

[21] O. Novo, “Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018, doi: 10.1109/JIOT.2018.2812239.

[22] Z. Bao, W. Shi, D. He, and K. K. R. Choo, “IoTChain: A three-tier blockchain-based IoT security architecture,” *arXiv*, pp. 1–24, 2018.

[23] A. Outchakoucht, H. Es-Samaali, and J. P. Leroy, "Dynamic Access Control Policy based on Blockchain and Machine Learning for the Internet of Things," *International Journal of Advanced Computer Science and Applications*, vol. 8, no.7, pp. 417 - 242, 2017.

[24] Y. Liu, F. R. Yu, X. Li, H. Ji and V. C. M. Leung, "Blockchain and Machine Learning for Communications and Networking Systems," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1392-1431, 2020, doi: 10.1109/COMST.2020.2975911.

[25] N. Haq, A. Onik, Md. Hridoy, M. Rafni, F. Shah and D. Farid, "Application of Machine Learning Approaches in Intrusion Detection System: A Survey" *International Journal of Advanced Research in Artificial Intelligence (IJARAI)*, vol. 4, no. 3, pp 9-18, 2015. http://dx.doi.org/10.14569/IJARAI.2015.040302

[26] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey". *Applied Sciences*. vol. 9, no. 20, 4396. 2019.

[27] J. Lever, M. Krzywinski, and N. Altman, "Logistic regression," Nat. Publ. Gr., vol. 13, no. 7, pp. 541–542, 2016, doi: 10.1038/nmeth.3904.

[28] Q. Cheng, P. K. Varshney and M. K. Arora, "Logistic Regression for Feature Selection and Soft Classification of Remote Sensing Data," in *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 491-494, Oct. 2006, doi: 10.1109/LGRS.2006.877949.

[29] P. Ranganathan, C. Pramesh, and R. Aggarwal, "Common pitfalls in statistical analysis: Logistic regression," *Perspectives in Clinical Research*, vol. 8, no. 3, pp. 148–151, Jul. 2017, doi: 10.4103/picr.PICR_87_17.

[30] J. Dou *et al.*, "Assessment of advanced random forest and decision tree algorithms for modeling rainfall-induced landslide susceptibility in the Izu-Oshima Volcanic Island, Japan," *Science of the Total Environment,* vol. 662, no. January, pp. 332–346, 2019, doi: 10.1016/j.scitotenv.2019.01.221.

[31] N. Gayatri, S. Nickolas, and A. V Reddy, "Feature Selection Using Decision Tree Induction in Class level Metrics Dataset for Software Defect Predictions," *Lecture Notes in Computational Science and Engineering*, vol. 2186, no. 1, pp. 124–129, 2010.

[32] "IoT network intrusion dataset | IEEE DataPort." [Online]. Available: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset. [Accessed: 03-June-2021].

[33] S. S. Alduayj and K. Rajpoot, "Predicting Employee Attrition using Machine Learning," *Proc. of the 13th International Conference on Innovations in Information Technology IIT 2018*, pp. 93–98, 2019, doi: 10.1109/INNOVATIONS.2018.8605976.

[34] S. Weber and J. Luo, "What makes an open source code popular on Git Hub?," *IEEE International Conference on Data Mining Workshops ICDMW*, vol. 2015-January, no. January, pp. 851–855, 2015, doi: 10.1109/ICDMW.2014.55.

[35] E. M. Geldiev, N. V. Nenkov, and M. M. Petrova, "Exercise of Machine Learning Using Some Python Tools and Techniques," *CBU International Conference Proceedings*, vol. 6, pp. 1062–1070, 2018, doi: 10.12955/cbup.v6.1295.

[36] C. F. Dormann *et al.*, "Collinearity : a review of methods to deal with it and a simulation study evaluating their performance," *Ecography 36,*. February 2012, pp. 27–46, 2013, doi: 10.1111/j.1600-0587.2012.07348.x.

Rawan Adel Shahin is currently a master candidate of computer science at the King Abdullah II School for Information Technology of University of Jordan, Amman, Jordan. She is currently a head of E-government Unit at Vocational Training Corporation (VTC). She received her B.Sc in information technology from Al-Balqa Applied University. Her areas of interests include but not limited to e-government, security and machine learning. She has publications in international journals.

Khair Eddin Sabri is a professor in the Computer Science Department at The University of Jordan. He obtained his B.Sc. degree in Computer Science from the Applied Science University, Jordan in June 2001. He also received M.Sc. degree in Computer Science from The University of Jordan in January 2004 and a Ph.D. degree in Software Engineering from McMaster University, Ontario Canada in June 2010. He has been co-organiser, program committee member and referee of many international workshops and conferences. His main research interests are formal methods, security, and AI. Dr. Sabri has many publications in international journals and conferences.