An Intelligent Detection of Malicious Intrusions in IoT Based on Machine Learning and Deep Learning Techniques

Saman Iftikhar, Danish Khan, Daniah Al-Madani,
Khattab M. Ali Alheeti, Kiran Fatima

Abstract

The devices of the Internet of Things (IoT) are facing various types of attacks, and IoT applications present unique and new protection challenges. These security challenges in IoT must be addressed to avoid any potential attacks. Malicious intrusions in IoT devices are considered one of the most aspects required for IoT users in modern applications. Machine learning techniques are widely used for intelligent detection of malicious intrusions in IoT. This paper proposes an intelligent detection method of malicious intrusions in IoT systems that leverages effective classification of benign and malicious attacks. An ensemble approach combined with various machine learning algorithms and a deep learning technique, is used to detect anomalies and other malicious activities in IoT. For the consideration of the detection of malicious intrusions and anomalies in IoT devices, UNSW-NB15 dataset is used as one of the latest IoT datasets. In this research, malicious and normal intrusions in IoT devices are classified with the use of various models.

Keywords: Malicious Intrusions, Anomaly detection, Machine Learning, Deep Learning, Classification, IoT dataset.

1 Introduction

The practice of integrating technologies such as sensors and software in everyday objects is on the rise. The idea is to enhance automation and

doi:10.56415/csjm.v30.16

enable the transfer of data without the need for computer-human interaction. Devices such as coffee machines, stoves, dryers, washers, and refrigerators contain smart capabilities, which stretch their functionality and user experience. In 2023, more than 18 million IoT devices will be connected to the Internet [1] (as shown in Fig. 1).

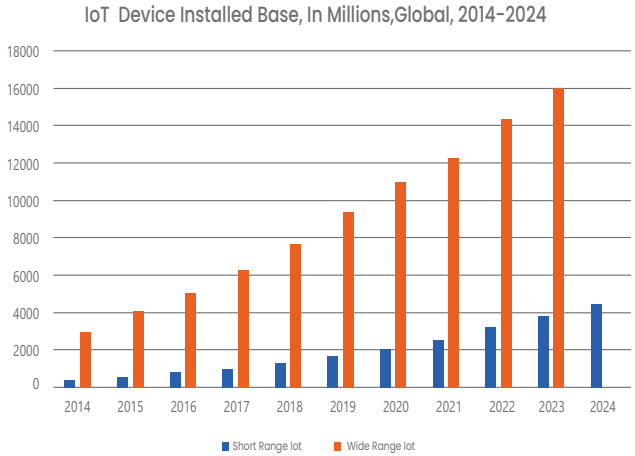

Figure 1. IoT devices installations [1]

At the same time, cybersecurity threats associated with IoT devices are on the increase. Top IoT threats are ranked according to their proportion rate (as shown in Fig. 2).

The occurrence of successful attacks targeting IoT systems can present dire consequences considering that these devices collect and store sensitive information. In addition to personal information, these devices contain sensors that take private images. From a legal and regulatory perspective, the need to ensure data privacy and confidentiality is paramount [3]. Machine learning, which encompasses machines, performs tasks without explicit programming, offers a promising way of detecting malware. Anomaly-based methods model the typical network behavior and identify abnormalities as plausible malware. This approach is advantageous as it offers a fairly effective method for dealing with both known and novel attacks [2]. The final hybrid approach

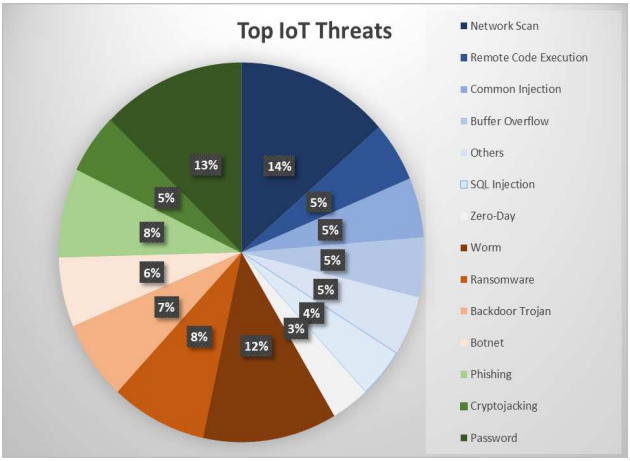

Figure 2. IoT Cybersecurity Threats [1]

combines both signature-based and anomaly-based methods. Machine learning techniques are routinely utilized to model the network behavior and identify anomalies. However, due to the high false positives, they often adopt the hybrid approach. By combining known malware signatures and network behavior, the ability of the intrusion detection system to detect malware accurately increases.

The application of deep learning in diverse big data fields has been successful due to the enhancement of computer processing power. Additionally, it offers an effective way of detecting attacks because of its high-level feature extraction capability. Essentially, deep learning uses a cascade of layers of processing units for extracting features from data, and each layer makes use of the output obtained from the prior layer as the input. Deep learning algorithms can adopt either supervised or unsupervised approaches. There seems to be an increasing adoption of deep learning methods as they can identify patterns and trends easily, can handle multi-variety and multi-dimensional data, and support continuous improvement. This paper intends to present an intelligent detection method for IoT systems that leverages effective classification, combined with various machine learning algorithms and a deep learning

algorithm, to detect benign and malicious activities.

The remainder of this paper is structured as follows: Section 2 presents the overview of related work. Dataset, Problem Statement and Proposed Framework is presented in Section 3. Subsections in Section 3 present more details of all the ML and DL algorithms that are used in this research. Section 4 presents the results of the proposed technique for intelligent detection of malicious activities in IoT networks. Section 5 details the conclusion of this research.

2 Related works

Different machine learning algorithms have been applied to anomaly detection in IoT systems and achieved positive outcomes [8], [9], [10], [11], [15], [16], [17], [19, 20, 21]. One of the most common approaches is the deep learning algorithm [8], [9].

Various datasets are available for examining the effectiveness of machine learning algorithms for detecting IoT-related intrusions [4], [5], [6], [22]. Koroniotis et al. developed one such dataset for the purposes of training and validating system credibility. The authors created a dataset called Bot-IoT, which integrated both simulated and legitimate internet of things traffic, including the different types of attacks. The study also encompassed presenting a test-bed environment for examining the current drawbacks of datasets, including capturing complete information about the network, accurate labeling, and emerging complex attacks. An evaluation of the BoT-IoT dataset using diverse machine learning and statistical methods compared with other datasets established adequate reliability [4].

Ullah and Mahmoud exploited a Botnet dataset from an existing one for detecting anomalous activity in IoT networks. The dataset had broader network and flow-based features, which were tested using diverse machine learning approaches, such as feature correlation, and recursive feature elimination. In addition to possessing the required accuracy levels, the dataset offers a good ground for analyzing anomalous activity detection models for IoT systems [5].

Sharafaldin et al. worked with an intrusion detection dataset. According to the researchers, most of the current datasets lack reliabil-

ity, especially in the face of emerging threats. After evaluating their dataset, the authors found that it exhibits reliability and accuracy when used together with machine learning algorithms to detect diverse attack categories [6]. However, this study did not focus on IoT-based systems, which is a key weakness as IoT networks tend to face unique attack threats. Nevertheless, the public availability of such datasets is imperative as it supports the creation and evaluation of IoT malicious detection models.

Shafiq et al. provided a specific detection method for malicious traffic in IoT systems. Despite the success of this study, the authors established that their approach did not detect some attacks accurately, especially keylogging. The performance of their approach for some machine learning algorithms such as decision trees and the random forest was also inadequate [1].

Dutta et al. followed the principle of stacked generalization to create an ensemble method that leverages deep learning models and a meta-classifier, which is the logistic regression [12]. The deep learning methods adopted encompassed the long short-term memory (LSTM) and the deep neural network (DNN) [12]. The approach utilized, encompassed two stages: the utilization of a Deep Sparse Auto-Encoder (DSAE) for feature engineering and a stacking ensemble for classification [12]. The evaluation of this approach showed that it is accurate in detecting network anomalies as compared to other state-of-the-art approaches. Abdullah et al. also illustrated the application of an ensemble method in the detection of network anomalies [13]. The system was based on dividing the input into different subsets based on the attack in question. Liu et al. presented a semi-supervised dynamic ensemble for detecting anomalies in IoT environments [14]. The algorithm combined mutual information criteria and semi-supervised extreme learning machine [14]. Experiments conducted on practical datasets showed that the proposed algorithm outperformed selected state-of-the-art approaches in terms of classification accuracy [14]. Evidently, ensemble approaches appear to be a promising direction of research mainly because of their ability to minimize biases and increase accuracy.

Tian et al. developed a distributed deep learning system for detecting web attacks on edge devices [8]. Based on the findings, the authors

established that deep learning is more effective in attack detection as compared to other approaches, especially when implemented in a distributed environment. Meidan et al. implemented deep encoders to detect IoT Botnet attacks [9]. The method entailed creating an algorithm that extracts behavior snapshots of the network and utilizes deep auto-encoders to detect abnormal traffic. The evaluation results showed that the method was effective in detecting attacks deployed using two widely known IoT-based Botnets. Thamilarasu and Chawla utilized a deep-learning algorithm to detect malicious traffic in IoT networks [11]. The system comprised network connection, anomaly detection, and mitigation modules. According to the authors, this system provided security as it served and facilitated interoperability between diverse network communication protocols utilized in IoT. The evaluation of the system demonstrated effectiveness and efficiency in detecting practical intrusions. Dutta et al. also illustrated the application of deep learning algorithms in the detection of anomalies in IoT systems [12]. Alhakami et al. implemented a non-parametric Bayesian approach to detect anomalies and identify intrusions [18]. This algorithm learned patterns of activities through a Bayesian-based MCMC inference for infinite bounded generalized Gaussian mixture models. The algorithm was tested and the outcomes showed that it was accurate in detecting diverse attacks.

More importantly, after detecting traffic anomalies, a system for classifying the attack is required. This classification is essential as it enables the implementation of the right controls. As a result, a more effective method is needed, which is the focus of this research. The idea is to adopt the ensemble approach of combining two or more machine learning methods to improve the accuracy for IoT attack detection.

3 Proposed Methodology

The application of machine learning algorithms to the detection of network anomalies and help in identifying and dealing with attacks is a recurrent topic in research literature [3]. Machine learning techniques developed often comprise two steps: training and testing [3]. Accordingly, the initial step encompasses identifying features or class attributes from

the training data. Afterwards, one has to identify a subset of attributed required to classify traffic either as normal or abnormal. This process is referred to as dimensionality reduction [3]. Once this is done, the model is trained using the training data and, thereafter, utilized in the classification of unknown data. During anomaly detection, the normal traffic pattern must be defined during the training process. Accordingly, when testing, the trained model is applied to new data, and every exemplar is classified as either anomalous or normal [3]. The same methodology has been followed to propose the ensemble approach in this paper and a step-by-step process is given below. According to Hasan et al., some of the attack and anomaly detection algorithms applied in IoT systems are logistic regression, decision trees, support vector machines, random forest, and artificial neural networks [7].

The growing research direction in developing machine learning systems for attack detection entails the utilization of ensemble algorithms [12, 13, 14]. The rationale behind ensemble learning is that a combination of two or more algorithms can produce better outcomes as compared to utilizing one algorithm alone. Stacking, bagging, and boosting are some of the common ensemble methods. Ensemble algorithms are ideal for classification as they minimize variance and biases hence boosting the accuracy of detection. In addition to whether it is supervised, semi-supervised, or unsupervised, the selection of an algorithm should also be based on its accuracy, recall, and precision.

3.1 Dataset to be used

The dataset used in the proposed study is known as UNSW-NB15 [4]. It is comprised of nine distinct types of attacks named as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. It contains 11 attributes named as start time, last time, attack category, attack subcategory, protocol, source address, source port, a destination address, destination port, attack name, and attack reference. Training set of the data includes $175,341$ records while the testing set contains $82,332$ records. The dataset is also comprised of various network attributes such as protocol (proto), state, sent packets (spkts), discarded packets (dpkts) and attack category (attack cat), etc. The labeling of the dataset is done based on the attack cat-

egories (*attack_cat*). The target label attack category depicts either 0 or 1 depending on whether the record is normal or attacked.

3.2 Problem Statement

The problem is to identify which network, network nodes, protocols, services and states are prone to intruder attacks. It's a general classification problem where different feature variables are classified on the basis of attacks faced by the network, and upon accurate classification, future attacks can be prevented by making that particular feature node category more secure.

3.3 Proposed Framework

First of all, the training and testing sets are read into the workspace. To make the data ready for model implementation, it is passed through several preprocessing steps. Algorithm 1 given below describes all the steps involved in the proposed framework. The architecture of the proposed framework is shown in Fig. 3.

Algorithm 1.
Input: $(proto)$, $(spkts)$, $(dpkts)$, $(attack\_cat) \in F = FeatureSet$
Output: classification of normal or malicious intrusions
Initialisation:
step 1: training set and testing set are concatenated
step 2: missing values from the dataset are removed or replaced
step 3: categorical feature $(attack\_cat)$ *selected*
step 4: $(attack\_cat)$ *is passed through "hot encoding" and "one hot encoding"*
step 5: all data values are normalized
step 6: dataset is split into training and testing data
step 7: parameter tuninigs are done as given below:
Logistic Regression (Log. Reg.): c *is a set equal to* 0.1. $c = array\ of\ c$ *that maps to the best scores across every class. If refit is set to false, then for each class, the best* c *is the average of the* $c's$ *that correspond to the best scores for each fold.* $C\_$ *is of shape(n_classes) when the problem is binary. Passive Aggressive Classifier (PAC): max iterations are set to 50*
K-Nearest Neighbor (KNN): n_neighbors are set to 3

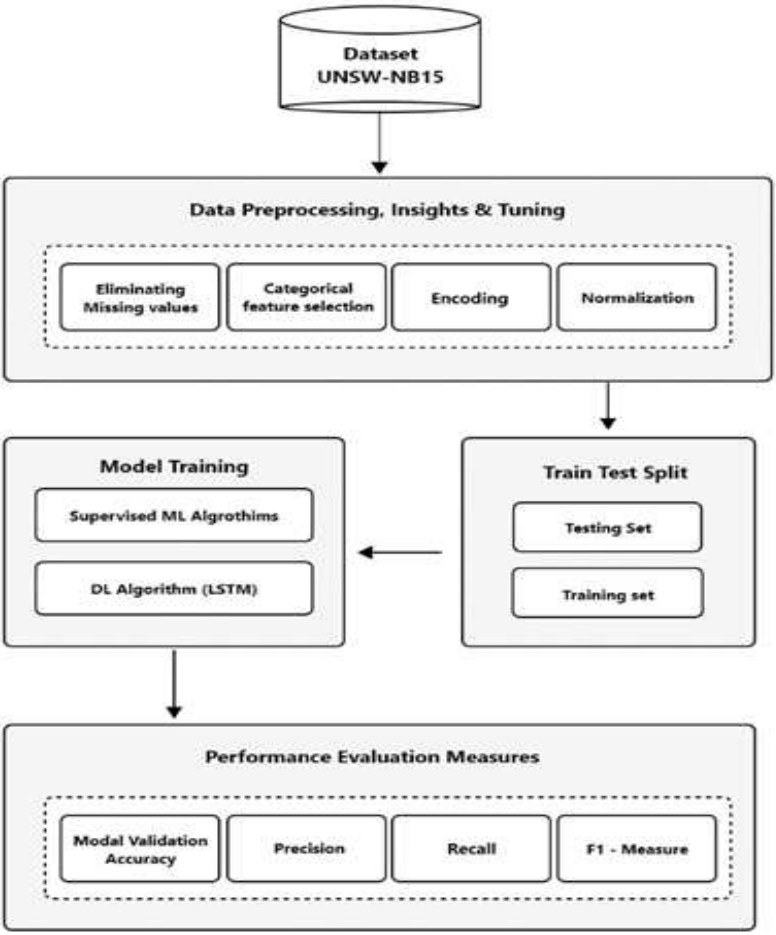

Figure 3. Architecture of Proposed Framework

Decision Tree Classifier (DT): random state is set to 1
Random Forest Classifier (Gini) (RF_Gini): n_estimators is set to 10
Random Forest (Information Gain) (RF_IFG): n_estimators is set to 10 and criterion = entropy
Other algorithms are applied as default
end of algorithm

3.4 Preprocessing and Data Preparation

Various preprocessing steps are applied so that the model can perform correctly on the dataset. Firstly, the missing values from the dataset are eliminated, then the dataset insights are taken in order to eliminate features that are not required or does not make any impact on the training of the model. Secondly, both the training and testing sets are concatenated for preprocessing process. Preprocessing can be done separately on training and testing partitions but it will have to be done twice which is why both the sets are concatenated. Finally, data normalization is performed to avoid overfitting.

3.5 Missing values

After the data have been concatenated, exploration is done to discover any missing values and then those values are replaced. The vast majority of the machine learning models that you want to employ will give you an error when you feed them NaN numbers. The best way is to avoid that and simply fill them with $0s$.

3.6 Data insights and tuning

Although, the dataset is clean but it still needs further processing in terms of One-hot-encoding for categorical data. E.g., "service" consists of different types, we have ftp, http, and '-' denoting not available or None. So we will need to treat it as a missing value as we will change it from '-' to 'None' instead of dropping the whole column. Unnecessary features, e.g., "id" need to be removed as well.

3.7 Categorical features selection and encoding

The attack categories also known as attack cats in the dataset, are considered as a categorical feature in the proposed study because these attack categories contain all kinds of attacks that are experienced by

the network traffic. One hot encoding is applied on the feature attack category values so that these values can be inputted into the model as target values.

3.8 Data Normalization

The dataset normalization is done in order to avoid overfitting. MinMaxScaler function is used to make sure that all the data values are between 0 and 1. As dataset size is large, so the MinMaxScaler function fits best in the scenario to normalize data. The normalization of the dataset will help machine learning algorithms to perform better or come to a conclusion faster. This is the reason, why it is a good practice to be followed before inputting such kind of data to the machine learning models.

3.9 Train and test splitting

The dataset is split into training and testing partitions in this case because our aim is to care for precision. For experimentation test data size is 30%. Parameter tunings are done and details about the use of various machine learning algorithms are given in Algorithm 1 given above. Some other machine learning algorithms such as Multi Nominal Naive Bayes (MNB), Gaussian Naive Bayes (GNB), Gradient Boosting (GB), Support Vector Machine (SVM) and deep learning algorithm (LSTM) are applied as default. The parameters' configuration of the ML and DL models is discussed below.

In the ML model MNB, alpha is set to 1.0, fit prior is set to true and class prior is set to none. Alpha is the parameter for additive smoothing of the data, and fit prior is used to whether learn or not learn about the prior class probabilities. We set class prior to none so that data prior class probabilities cannot be changed.

In the ML model GNB, the parameter prior is set to none. Variable smoothing parameter is also set so that zero probability can be handled. In the GB classifier, the parameters loss, learning rate equal to 0.1, and the number of estimators equal to 100 are set.

In the ML model SVM, the following parameters are set: parameter C equal to 1.0, kernel equal to rbf, gamma equal to scale, coefficient equal to 0.0, and shrinking equal to true. C is the parameter for regularization. The strength of the regularization goes down as C goes up.

The kernel is used to tell the algorithm what kind of kernel to use.

In the DL algorithm LSTM, batch size is set to 10, number of epochs are set to 100 with a validation split of 10 percent.

4 Results and Discussion

Finally, results are achieved through various machine learning models such as Logistic Regression, K-Nearest Neighbor (Lazy Algorithm), Decision Trees, Random Forest (Gini), Passive aggressive classifier, Multinominal naive bayes, Gaussian Naive Bayes, Gradient Boosting, SVM, Random Forest (Entropy or Information-gain), and a deep learning model such as LSTM.

The classification phase focuses on achieving best accuracy. Two types of accuracies are considered here, training accuracy and Cross-Validation (CV) accuracy. Table 1 shows training accuracies achieved by implementing the above-mentioned ML algorithm and LSTM that is a deep learning algorithm. The table also shows the execution time taken by each algorithm, validation accuracies, precision, recall, and F1 score.

It can be seen above that for our proposed method "Random Forest (Information Gain)" and "Passive Aggressive Classifier" provides the best cross-validation accuracies. These two machine learning algorithms have outperformed in the intelligent detection of benign and malicious attacks in the concerned IoT dataset used in our proposed detection method. Validation accuracies, precision, recall and F1 score for each algorithm used for classification of anomalies are explained in (equation 1, 2, 3, and 4) respectively.

K-fold cross validation accuracy is utilized to compute the accuracies of the models used in this study. 10 folds of the dataset are used to avoid overfitting.

$$Accuracy = \frac{[TN] + [TP]}{[FN] + [FP] + [TN] + [TP]}, \tag{1}$$

where TP, TN, FP and FN stands for true positive, true negative, false positive, and false negative, respectively. TP is originally True as well as predicted True, TN is originally True, but predicted negatively

Table 1. Performance measures of the models

Model	Accuracy %	Execution Time	CV Accuracy	Precision	Recalls	F1–score
Logistic Regression	90.1	93.89	8.72	0.90	0.90	0.89
Decision Tree	99.74	93.89	91.1	0.99	0.99	0.99
Random Forest (Gini)	99.74	325.76	92.15	0.99	0.99	0.99
Random Forest (Information Gain)	99.74	388.32	92.2	0.99	0.99	0.99
Passive Aggressive Classifier	85.77	28.48	87.11	0.87	0.85	0.85
KNN	95.92	4040.04	89.71	0.95	0.95	0.95
Multi Nominal Naïve Bayes	74.45	15.24	74.33	0.80	0.74	0.74
Gaussian Naïve Bayes	50.5	12.12	50.46	0.79	0.50	0.44
Gradient Boosting	93.38	1234.75	91.57	0.93	0.93	0.93
SVM	90.28	367.53	88.78	0.90	0.90	0.90
LSTM	93 on 20 epochs	139.39	0.9359	0.9664	–	–

by the classifier. Figure 4 shows the k-fold accuracy of DL and ML models.

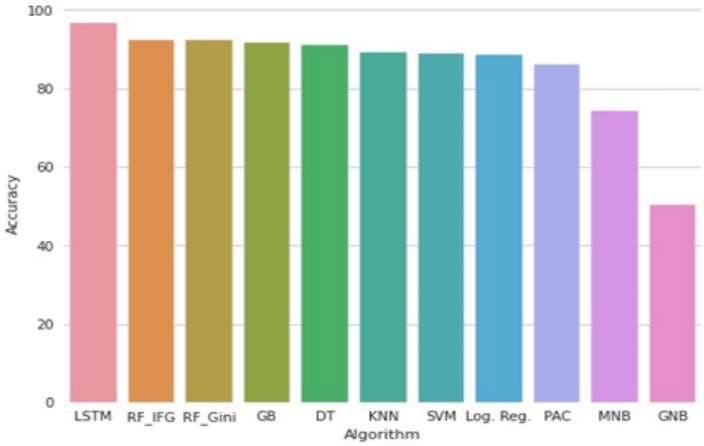

Figure 4. Validation Accuracies of various algorithms applied in the proposed study

Precision is percentage of correct predictions of a class among all predictions for that class.

$$Precision = \frac{[TP]}{[FP] + [TP]}, \tag{2}$$

where TP stands for true positive and FP stands for false positive. FP depicts the record which is originally false but predicted positive by the classifier. Figure 5 depicts precision score achieved by various machine and deep learning algorithms.

Recall is proportion of correct predictions of a class and the total number of occurrences of that class.

$$Recall = \frac{[TP]}{[FN] + [TP]}, \tag{3}$$

where TP stands for true positive and FN stands for false negative. FN depicts the record which is originally false and predicted false by the

classifier. In Fig. 6, precisions of various machine and deep learning algorithms are shown.

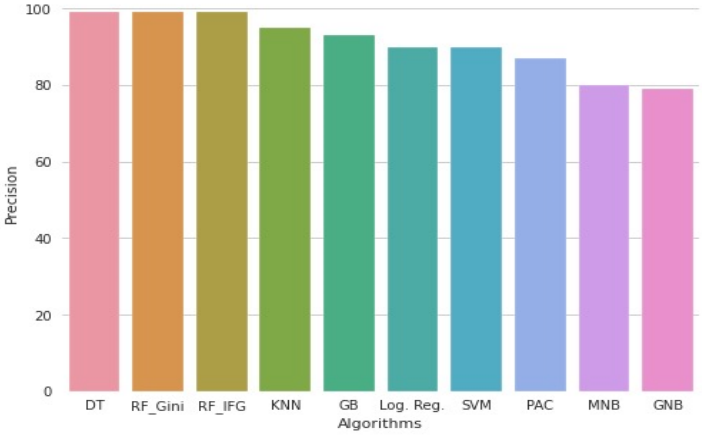

Figure 5. Precision of various algorithms applied in the proposed study

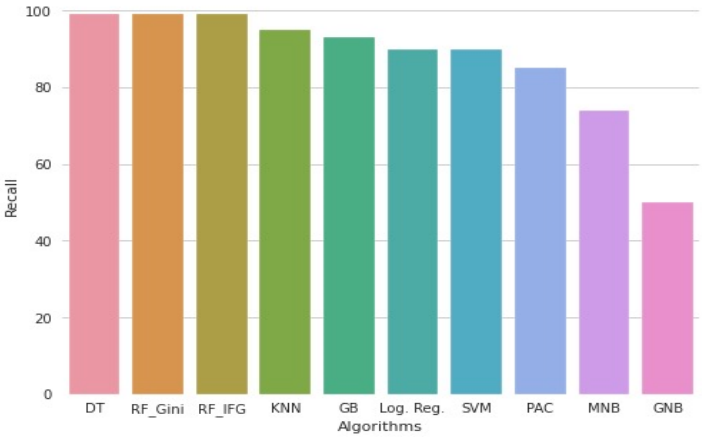

Figure 6. Recall of various algorithms applied in the proposed study

F1-score is a single metric combination of precision and recall.

$$F1 - score = 2 * \frac{[Recall] * [Precision]}{[Recall] + [Precision]}. \quad (4)$$

Figure 7 depicts the F1-score achieved by machine and deep learning algorithms.

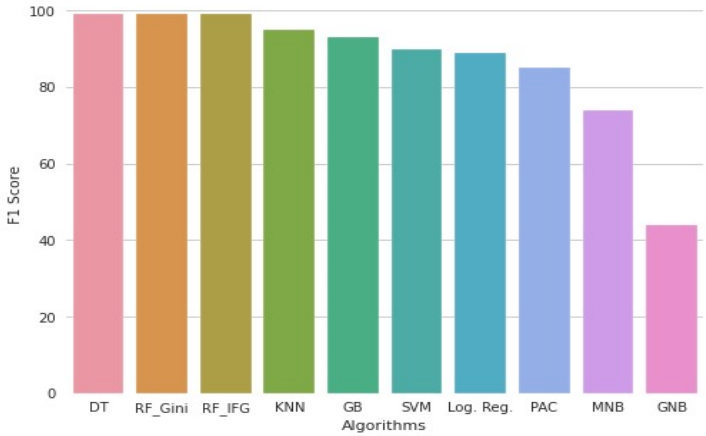

Figure 7. F1-score of various algorithms applied in the proposed study

For the evaluation, our proposed approach has compared with the state-of-art technique provided in [1] for training accuracy, execution time, Cross-Validation (CV) accuracy, precision, recall, and F1 score. Results are outstanding in the case of our proposed detection method and are shown in Fig. 8.

5 Conclusions

To classify benign and malicious intrusions in IoT devices is the big issue in this age of internet. In this research, Machine Learning algorithms and a Deep Learning method are being utilized for the intelligent detection of anomalies or malicious activities in IoT. Among many features being controlled during IoT network traffic, only some of them are responsible for the activation of malicious activities. This paper has

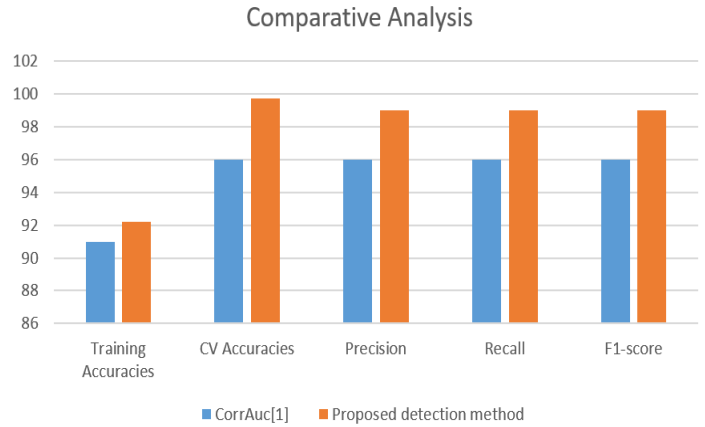

Figure 8. Comparative Analysis of CorrrAuc [1] and Proposed Approach

proposed an intelligent malicious detection technique that is basically an ensemble approach combined with various machine learning algorithms and deep learning algorithms. Apart from various algorithms being used in the experimentation, "Random Forest Information Gain" and "Passive aggressive classifier" provided the best cross-validation accuracies for anomaly detection. Moreover, after the evaluation, our proposed detection method has shown the best accuracies around 99%, which is more than any other state-of-art techniques developed in the literature.

Acknowledgement

The authors would like to thank Arab Open University, Saudi Arabia for supporting this study.

References

[1] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques," *IEEE Internet of*

Things Journal, vol. 8, no. 5, pp. 3242–3254, March 1, 2021, DOI: 10.1109/JIOT.2020.3002255.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[3] R. M. Alhajri, A. B. Faisal, and R. Zagrouba, "Survey for anomaly detection of IoT botnets using machine learning auto-encoders," *Int J Appl Eng Res*, vol. 14, no. 10, pp. 2417–2421, 2019.

[4] N. Koroniotis et al., "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-Iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[5] I. Ullah and Q. H. Mahmoud, "A Technique for Generating a Botnet Dataset for Anomalous Activity Detection in IoT Networks," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 134–140, 2020.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pp. 108–116, 2018.

[7] M. Hasan et al., "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, Article ID: 100059, 2019. DOI: 10.1016/j.iot.2019.100059.

[8] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[9] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "A network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12-–22, 2018.

[10] V. Timcenko and S. Gajin, "Machine learning based network anomaly detection for IoT environments," in *ICIST-2018 Conference*, 2018.

[11] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the internet of things," *Sensors*, vol. 19, no. 9, pp. 1977, 2019. DOI: https://doi.org/10.3390/s19091977.

[12] V. Dutta et al., "A deep learning ensemble for network anomaly and cyber-attack detection," *Sensors*, vol. 20, no. 16, pp. 4583, 2020.

[13] M. Abdullah et al., "Enhanced intrusion detection system using feature selection method and ensemble learning algorithms," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 16, no. 2, pp. 48–55, 2018.

[14] S. Liu et al., "A semi-supervised dynamic ensemble algorithm for IoT anomaly detection," in *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pp. 264–269, 2020.

[15] S. Egea, A. R. Manez, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Intelligent IoT traffic classification using novel search strategy for fast based-correlation feature selection in industrial environments," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1616—1624, 2018.

[16] H. Zhang, G. Lu, M. T. Qassrawi, Y. Zhang, and X. Yu, "Feature selection for optimizing traffic classification," *Computer Communications*, vol. 35, no. 12, pp. 1457—1471, 2012.

[17] S. Su et al., "A correlation-change based feature selection method for IoT equipment anomaly detection," *Applied Sciences*, vol. 9, no. 3, pp. 437, 2019.

[18] W. Alhakami et al., "Network anomaly intrusion detection using a nonparametric bayesian approach and feature selection," *IEEE Access*, vol. 7, pp. 52181–5219, 2019.

[19] M. Shafiq et al., "Effective feature selection for 5G IM applications traffic classification," *Mobile Information Systems*, vol. 2017, Article ID: 6805056, 12 pages, 2017.

[20] I. H. Sarker et al., "Intrudtree: A machine learning based cyber security intrusion detection model," *Symmetry*, vol. 12, no. 5, Article No. 754, 15 pages, 2020.

[21] I. Ullah and Q. H. Mahmoud, "A two-level flow-based anomalous activity detection system for IoT networks," *Electronics*, vol. 9, no. 3, Article No. 530, 2020. DOI: https://doi.org/10.3390/electronics9030530.

[22] R. Ahmad and I. Alsmadi, "Machine learning approaches to IoT security: A systematic literature review," *Internet of Things*, Article ID: 100365, 2021.

Saman Iftikhar, Danish Khan, Daniah Al-Madani, Khattab M Ali Alheeti, Kiran Fatima

Received January 10, 2022
Revised July 24, 2022
Accepted September 1, 2022

Saman Iftikhar
Faculty of Computer Studies, Arab Open University, Saudi Arabia
E–mail: s.iftikhar@arabou.edu.sa

Danish Khan
Department of Computer Science, COMSATS University Islamabad
Wah Campus, Wah Cantt Pakistan
E–mail: danish56566@gmail.com

Daniah Al-Madani
Faculty of Computer Studies, Arab Open University, Saudi Arabia
E–mail: d.almadani@arabou.edu.sa

Khattab M Ali Alheeti
Computer Networking Systems Department,
College of Computer Sciences and Information Technology,
University of Anbar, Anbar, Iraq
E–mail: co.khattab.alheeti@uoanbar.edu.iq

Kiran Fatimah
TAFE, NSW Australia
E–mail: kiran.fatima4@tafensw.edu.au