Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 10, October 2020, pg.95 – 103

Attack and Anomaly Detection in IoT Networks using Machine Learning

Dr. R. Thamaraiselvi[1]; S. Anitha Selva Mary[2]

[1]Head, Department of Computer Applications, Bishop Heber College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, India
[2]Department of Computer Applications, Bishop Heber College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, India
[1] thams.shakthi@gmail.com; [2] anitavk1995@gmail.com

DOI: 10.47760/ijcsmc.2020.v09i10.012

Abstract: ***For quite a few years now the name Internet of Things (IoT) has been around. IoT is a technology capable of revolutionizing our way of life, in sectors ranging from transportation to health, from entertainment to our interactions with government. Even this great opportunity presents a number of critical obstacles. As we strive to develop policies, regulations, and governance that form this development without stifling creativity, the increase in the number of devices and the frequency of that increase presents problems to our security and freedom. This work attentions on the security aspect of IoT networks by examining the serviceability of machine learning algorithms in detecting anomalies that are contained within such network data. It discusses (Machine Learning (ML) algorithms which are used effectively in relatively similar situations and compares them using several parameters and methods. The following algorithms are implemented in this work: Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM), and Decision tree Algorithm. The Random Forest algorithm obtained the best results, with an accuracy of 99.5 per cent.***

Keywords: IoT, Machine Learning, Security

I. INTRODUCTION

The Internet of Things (IoT) is proclaimed as a creation that will bring about drastic changes in how we live. It is recognized as an enabler that will improve productivity in a change of fields, including transportation and logistics, healthcare and development. The IoT will assist in process optimization through advanced data analytics, and will be the catalyst for new business opportunities by trying to capitalize on its cyber-physical features, generating cross-cutting applications and services. The idea of linking 'things' to the internet goes far beyond using the word 'Internet of Things'. At Carnegie Melon University, students in the early 1980s fitted internet-connected photo sensors to a soft drink vending machine, enabling them to count the number of cans that were being dispensed. This allowed anyone with internet access to decide how many drinks had been dispensed, and hence how many were left.

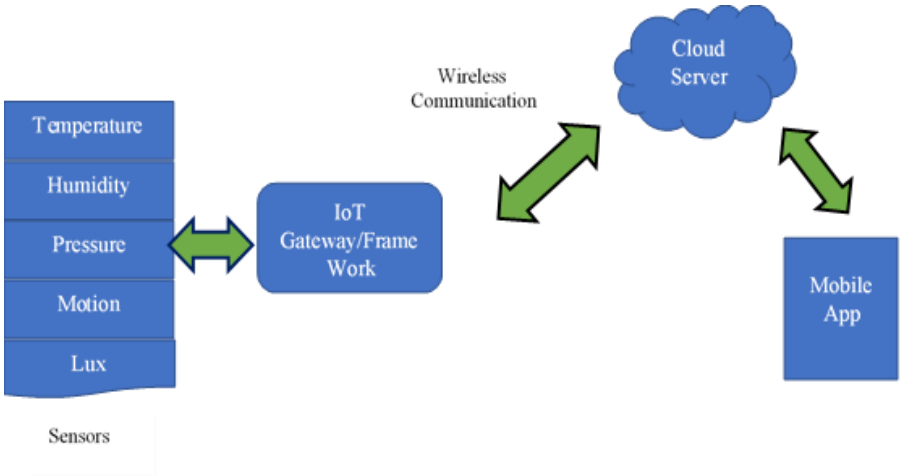

Figure 1. IOT Architecture

Currently, stability is the main concern for more than half of future IoT device users. Thus, researchers have begun to explore more innovative security mechanisms to tackle this problem in the last few years. There are two major security measures categories: passive (e.g. passwords, encryption), and active. One of those recent steps is the use of machine learning to identify and classify attacks,[1] because the two technologies seem to match each other well in concept. To build up its detection model, machine learning algorithms require huge amounts of data and IoT systems can provide them. Moreover, the sheer number of types of attacks and their manifestations makes it virtually impossible for human operators to classify and categorise them [1].The main objective of this paper is to create Machine Learning algorithms for use in network-based anomaly detection on the Internet of Things devices, and then test them using the IoT-23 dataset [6], a new dataset consisting of malicious and benign network captures from a variety of IoT devices.

II. RELATED WORK

Chatterjee et al. [2] proposed an ANN-based authentication mechanism in IoT networks. In essence, Physically Unclonable Function (PUF)-based authentication can be effective in IoT where the physical properties of the transmitter (communicating device) are analyzed (which are otherwise discarded as impurities of the communication). In the same spirit, Chatterjee et al. accumulated these nonidealities of the communication and applied in-situ ML algorithm to classify the transmitters. The ML algorithm is applied at the receiver end and entropy is extracted to successfully identify and/or classify the transmitters. The authors measured the inaccuracy in detection, or false detection of their scheme. With 4800 transmitters, the detection error was $< 10^{-3}$ and slightly increased (about 10^{-2}) with 10,000 transmitters. From efficiency standpoint, there is no overhead incurred by the transmitter whereas the receiver needs two neural networks that will incur about 3%-5% additional power.

Ferdowsi et al. [3] proposed a distributed GAN based intrusion detection scheme for IoT networks. Unlike the traditional GANs, the authors employed distributed GANs to avoid the communication cost incurred by the centralized GAN. Furthermore, in case of a centralized GAN, the access to all the data points from the IoT devices should be granted. Therefore, it could also compromise the privacy. Whereas in the distributed GAN, the data is not shared by the IoT devices, rather the weights are shared among the IoT nodes. The authors reported that the distributed GANs could achieve about 20% higher detection accuracy, 25% higher precision, and 60% lower false positive as compared to the centralized GAN.

Karbab et al. [4] proposed MalDozer, a DL-based malware analysis tool for android application framework. The detection framework is based on ANN and tested with both benign and malware applications in the android platform. In essence, MalDozer is based on sequences such as API method calls in android, resource permissions, and raw method calls. Furthermore, the proposed technique also automatically engineers features during training. The experimental results achieved detection rate of 96% to 98% android malware with the correct malware family.

Pajouh et al. [5] proposed an RNN-based DL approach for malware analysis technique in IoT. The authors considered Advanced RISC Machines (ARM)-based applications in IoT. The authors train their models with different existing malware datasets and then test their framework with the new malware. Through experiments, the authors concluded that LSTM classifiers deliver best results among other classifiers. More precisely, the RNN-based DL technique achieved 98% accuracy in detecting new malware inside IoT application. Similarly, in another work.

III. METHODS

This section of the paper concerns the collection of data, how it was pre-processed and theoretical discussions on the algorithms and the measurements used in this project. The first big step is preprocessing data, which consists of data collection, data analysis, data preparation, statistical inference and splitting of data. These steps processed the data so the algorithms could be fed in. The data was divided randomly in a ratio of 80-20, with the 20 percent being the training data, and the 80 percent being the test data. Many of the algorithms are of multi-class form. Finally, the algorithms on accuracy, the f1-score, the recall score and the support score were compared.

a) Dataset: The collection of data used in this project is IoT-23[6], a dataset generated by the Avast AIC Lab. The dataset includes 20 captures of malware from different IoT computers, and 3 captures for benign anomalies. The data were collected in collaboration with Prague's Czech Technical University, with data obtained between 2018 and 2019[6]. The full form of the dataset includes:.pcap files, the original network capture files, conn.log.labeled files, generated by consecutively the Zeek network analyzer, different descriptions and information about each capture..

TABLE 1. Dataset Description

Column	Description	Type
ts	The time when the capture was done, expressed the Unix true	int
uid	The ID of the capture	str
id_orig.h	the IP address where the attack happened, either IPv4 or IPv6	str
id_orig.p	The port used by the responder	str
Id_resp.h	The IP address of the device on which the capture happened	int
id_resp.p	the port used for the response from the device where the capture happened	int
proto	the network protocol used for the data package	Str
service	The application protocol	str
duration	the amount of time data was traded between the device and the attacker	float
orig_bytes	the amount of data sent to the device	int
resp_bytes	the amount of data sent by the device	int
conn_state	the state of the connection	str
local_orig	whether the connection originated locally	bool
local_resp	whether the response originated locally	bool
missed_bytes	number of missed bytes in a message	int
history	the history of the state of the connection	str

orig_pkts	number of packets being sent to the device	int
orig_ip_bytes	number of bytes being sent to the device	int
resp_pkts	number of packets being sent from the device	int
resp_ip_bytes	number of bytes being sent from the device	int
tunnel_parents	the id of the connection, if tunnelled	str
label	the type of capture, benign or malicious	str
detailed_label	if the capture is malicious, the type of capture, as described above	str

The conn-state column is a Zeek-specific variable, which represents the relation status between two devices. As an example, S0 means that a computer tries to communicate but the other side doesn't respond. In this dataset, all missing values from any of the entries were marked with a dash ("-"), with the exception of the IP address, which was marked with two colon("::").

b)Machine Learning Algorithms in IoT Security:Machine learning is the skill of having computer to learn and behave in an autonomous fashion as human do and improving their learning over time. Machine learning algorithm uses statistics to find patterns and learn from them in vast quantities of data, then make a decision or prediction about something.You can classify machine learning into three forms as in Fig. 3. The database on which we are training our model is marked in supervised learning. Input and output (1) are simply and distinctly mapped, where x variable is data, and y variable is output. The model is able to get trained based on the example data.

$f(x) \rightarrow y$ (1)….There are no labelled data in unsupervised learning, and the algorithm identifies and learns from the pattern inside the database. The unsupervised algorithm group the data into different clusters based on their density or properties.Reinforcement learning is evolving and the most common type of machine learning algorithm, with this algorithm attempting to achieve the target in a complex environment. It achieves the target based on system-supplied reward and penalty.There are several algorithms that can be used to protect IoT infrastructure in an efficient way. Supervised learning works best when we know the variable context, i.e. the output that corresponds to each input. We use unsupervised learning where we are not concerned with the output, primarily this learning is used to categorise the properties. Reinforcement learning, on the other hand, is distinct from two learning mechanisms, in which learning software agent learns from their own positive or negative experience.

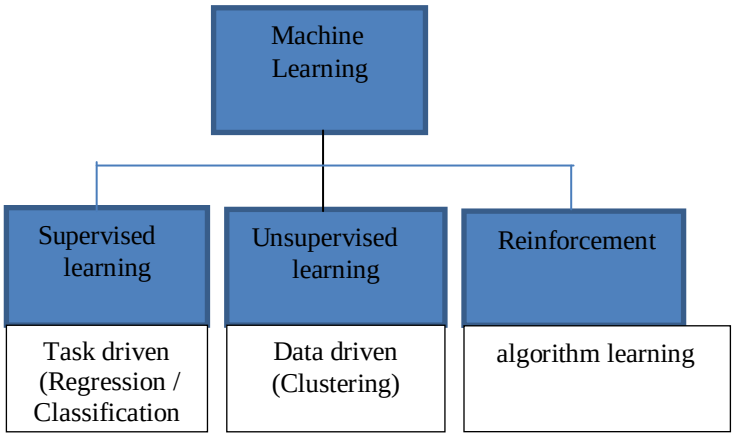

Figure 2. Machine Learning Fields

At Figure 3, One of the commonly supervised learning algorithms is support vector machine (SVM). It can be used both for the task of classification and regression. But it is often used in the issue of classification. In this algorithm we plot each data object as a point in n-dimensional space with the value of each function being the value of a particular coordinator, then we perform classification by finding the hyper plane that differentiates the two classes.

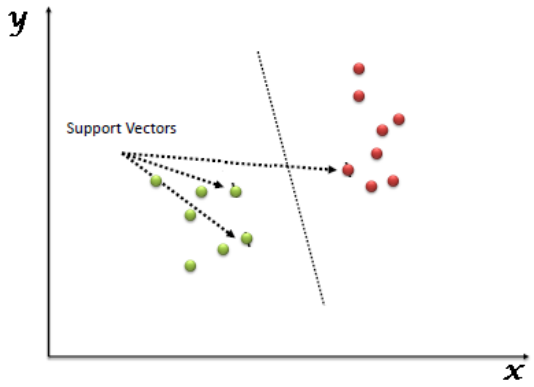

Figure 3. Support Vector Machine (Hyper plane differentiating two classes)

Other supervised learning algorithm is the random forest. There is a direct relationship between forest numbers of trees, the number of decision tree, the more accurate would be the outcome. Decision tree is a basic element of this algorithm and is also a support tool for decision making. A graph like a tree is used to demonstrate future effects. The principal benefit is that this algorithm can be used for the purpose of classification and regression.The classifier Naive Bayes is a simple and efficient algorithm for the classification task. When we use it for textual data analysis, like natural language processing, it provides great results. Based on the Bayes ' Theorem, which is based on conditional probability, Naive Bayes Classifiers implies an event A will occur, provided that another event B has already occurred. The theorem (2) allows updating of a hypothesis A every time new proof B is presented.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \qquad (2)$$

Where P is signified as probability, P (A | B) is the probability of incident A happening given that B has happened already. P (B | A) is the probability of the event B happening given that A has occurred. P (A) the probability of event B occurring and P (B) probability of event A occurring.

Decision trees are a classifier that attempts to go from data point information to conclusions about its importance within the given system by making a series of simple, successive decisions. Although decision trees are not very robust classifiers, they are commonly used as a vases for other, more complex classifiers, such as forests and boosters, owing to their being very computational light.

IV. ANALYSIS METHODS

In order to evaluate the algorithms defined above, the metrics presented below were used. The discussion of these metrics in the context of this project is done in the Results section. There are four concepts which have to be presented before the metrics are discussed:

TP is the number of actual positives that were correctly identified
TN is the number of actual negatives that were correctly identified
FP is the number of actual positives that were identified as negatives
FN is the number of actual negatives that were identified as positives

V. CONFUSION MATRIX

A confusion matrix is a table that allows for the conception of the presentation of a model by showing which values the model thought belong to which classes. It has a N ×N size, where n is the number of classes, with the columns representing the actual classes and the rows the predicted classes.

Table 1. Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

a)Precision

The precision is a metric that evaluates the model by calculating the fraction of correctly identified positives. Its formula is:

$$Precision = \frac{TP}{TP + FP}$$

b)Accuracy

The accuracy is a metric that evaluates the model by calculating the fraction of correct predictions over the total number of predictions. Its formula is:

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

c)Recall Score

The recall score is a metric that evaluates the model by calculating the fraction of actual positives that were correctly identified. Its formula is:

$$Recall = \frac{TP}{TP + FN}$$

VI. EXPERIMENTAL SETUP

This experiment was conducted on a machine operating system windows 10 Home 64 bit Build 18737, the processor was an Intel® Core (TM) i3 10th Generation CPU @ 3,62GHz. The memory of Machine was 6GB RAM. The graphics card was NAVIDIA 6GB. The algorithm used for this research were implemented using Java 10 64 bit, loading the data into the models was done using Swing library.

VII. RESULT ANALYSIS

As the data set was split into 2 parts of similar size, with the data being distributed randomly between them, the metrics obtained from each of the parts can sets were there major imbalance between the size of the class. The tool is designed to be generic to accept any datasets that needs to perform classification analysis. The classification results obtained by the algorithms proposed in the architecture are shown in Table 2.

Table 2. Classification Results of learning Algorithms

Performance Measures	**NB**	**Decision tree**	**SVM**	**Random Forest**
Accuracy	78.8406%	96.3%	99.4%	99.5%
True Positive	0.788	0.963	0.994	0.995
False Positive	0.217	0.963	0.014	0

Precision	0.788	0.927	0.994	1
Recall	0.788	0.963	0.994	0.995
F-Measure	0.788	0.945	0.99	0.99

Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

Figure 4. specifies the accuracy analysis of the ML algorithm with different ML algorithms as the motivation of the work is to identify the best features that increase the accurate prediction of IoT Dataset. Among the three experimental algorithms, Random Forest Algorithm improves the accuracy of other in Iot dataset prediction with 99.5%.

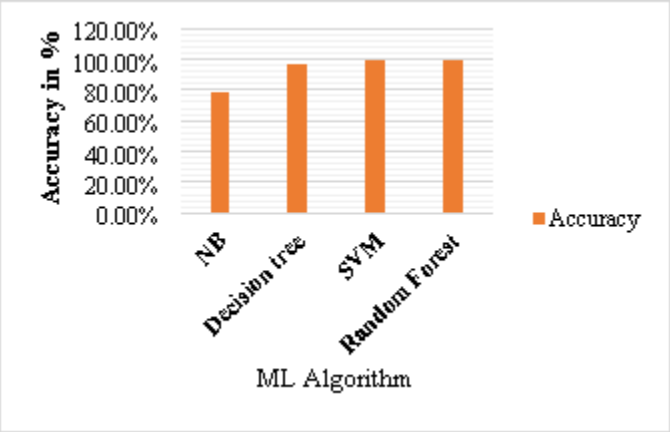

Figure 4. Accuracy Analysis of learning algorithms

Figure 5. depicts the precision value analysis of experimental algorithms for IoT detection. Random Forest algorithm has the highest precision value than other algorithms.

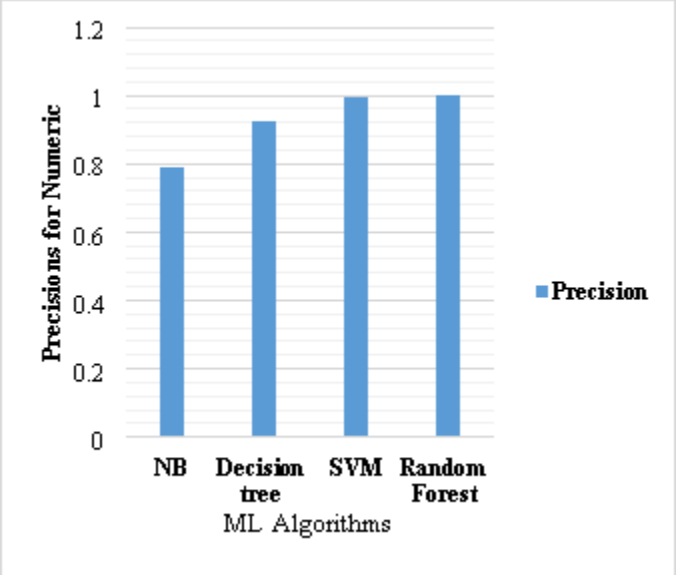

Figure 5. Precision Measure

Recall is a measure that results how many relevant IoT detections are selected. Recall measure analysis shown in Fig.6. Proves that the algorithm New proposed algorithm with SVM outperforms than the other algorithms.

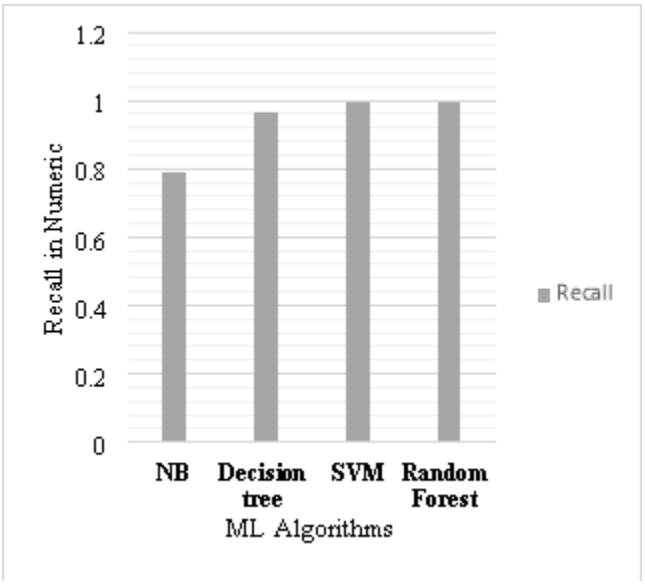

Figure 6. Recall Measure

Precision measures quality and recall measures quantity. F-measure is the harmonic mean of precision and recall measures as denoted in.

$$F - Measure = 2 \times \frac{Precision.Recall}{precision + recall}$$

F-Measure analysis of proposed algorithms is shown in Figure .7 Like other results, the F-Measure value of Random Forest with SVM is high. The results denote that the algorithm outperforms the experimental algorithms in terms of quality and quantity.

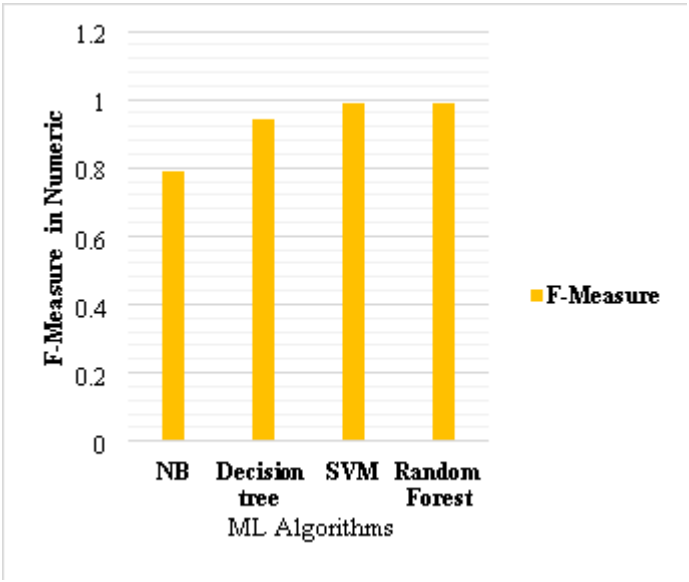

Figure 7. F-Measure Analysis

In this work Machine Learning algorithm is used. The results of the work show that the algorithm proposed extracts a minimal number of characteristics that improve prediction accuracy by 99.5% misclassifying one single instance in the IoT data set for the Anomaly detection.

VIII. CONCLUSIONS

In conclusion, the Random Forest algorithm is the best choice for anomaly detection and classification in the context of the IoT-23 dataset. While the reasons behind this conclusion are not fully understood, this algorithm scored the highest in all metrics and it presented itself as the best choice overall. To answer the secondary research question, the results of this study are in line with what other similar works found as well. The future research is to find out what the minimum amount of data from the IoT-23 data set is such that the implemented models are still accurate. Also, as mentioned in the Result Analysis section, the reason behind the high accuracy of the SVM classifier should be investigated. Possible future research would be to use more advanced types of Artificial Neural Networks and see whether they get different results.

REFERENCES

[1]. Zeadally, Sherali, and Michail Tsikerdekis. "Securing Internet of Things (IoT) with machine learning." International Journal of Communication Systems 33, no. 1 , 2020.

[2]. Chatterjee, Baibhab, Debayan Das, Shovan Maity, and Shreyas Sen. "RF-PUF: Enhancing IoT security through authentication of wireless nodes using in-situ machine learning." IEEE Internet of Things Journal 6, no. 1 (388-398): 2018.

[3]. Ferdowsi, Aidin, and Walid Saad. "Generative adversarial networks for distributed intrusion detection in the internet of things." In 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1-6. IEEE, 2019.

[4]. Karbab, ElMouatez Billah, Mourad Debbabi, Abdelouahid Derhab, and Djedjiga.

[5]. Mouheb. "MalDozer:Automatic framework for android malware detection using deep learning." Digital Investigation 24 (S48-S49): 2019.

[6]. HaddadPajouh, Hamed, Ali Dehghantanha, Raouf Khayami, and Kim-Kwang Raymond Choo. "A deep recurrent neural network-based approach for internet of things malware threat hunting." Future Generation Computer Systems 85 (88-96): 2018.

[7]. Agustin Parmisano, Sebastian Garcia, M. J. E. IoT-23 Dataset: A labeled dataset of Malware and Benign IoT Traffic. — Stratosphere IPS, 2020.

[8]. Shafiq, Muhammad, Zhihong Tian, Yanbin Sun, Xiaojiang Du, and Mohsen Guizani. "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city." Future Generation Computer Systems 107 (433-442): 2020.

[9]. Hasan, Mahmudul, Md Milon Islam, Md Ishrak Islam Zarif, and M. M. A. Hashem. "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches." Internet of Things 7 (100059): 2019.

[10]. Hou, Jianwei, Leilei Qu, and Wenchang Shi. "A survey on internet of things security from data perspectives." Computer Networks 148 (295-306): 2019.

[11]. Mishra, Preeti, Vijay Varadharajan, Uday Tupakula, and Emmanuel S. Pilli. "A detailed investigation and analysis of using machine learning techniques for intrusion detection." IEEE Communications Surveys & Tutorials 21, no. 1 (686-728): 2019.

