IoT-Botnet Detection using Long Short-Term Memory Recurrent Neural Network

Jenny Costa
Computer Engineering
SRIEIT, Goa University
Shiroda, Goa

Nandini Fal Dessai
Computer Engineering
SRIEIT, Goa University
Shiroda, Goa

Shivani Gaonkar
Computer Engineering
SRIEIT, Goa University
Shiroda, Goa

Dr. Shailendra Aswale
Computer Engineering
SRIEIT, Goa University
Shiroda, Goa

Pratiksha Shetgaonkar
Computer Engineering
SRIEIT, Goa University
Shiroda, Goa

***Abstract*— IoT (Internet of Things) provides unique identifiers and the facility to convey data across a network without requiring human-to-computer or human-to-human interaction. It is one of the most fast-evolving technologies nowadays. The positive influence of the IoT toward governments, citizens, and businesses is being significant. IoT comes with significant security concerns that need to be addressed. One of the serious security threats in network security is IoT-Bot. In past years many techniques are being practiced to detect IoT-Bot in a network. This paper explains the detection of IoT-botnet using a deep learning-based LSTM RNN (Long Short-Term Memory Recurrent Neural Network) model. The accuracy of this model is then compared with the SVM (Support Vector Machine), LR (Linear Regression), and KNN (K-Nearest Neighbors) model. UNSW-NB15 dataset is used for training and testing the model. The dataset contains 9 types of attack categories. The accuracy of this proposed model is very high. This can further be extended to work in real-time botnet detention. Wireshark can be used to collect real-time network traffic and detect IoT-Bot in a network.**

Keywords*— IoT, Deep learning, LSTM RNN, IoT-Botnet, DoS (Denial-of-Service), Reconnaissance.

I. INTRODUCTION

Since the 1960s, the Internet has been playing a very important role in connecting individuals and putting businesses and organizations together. Its development has made our life fast and easy but at the same increased the challenges in dealing with its security and privacy. The IoT (Internet of Things) refers to a huge number of "things" that are connected to the internet so they can share data with other things (IoT applications, connected devices, industrial machines, and more). The devices which are connected to the internet use built-in sensors for data collection and, in some cases, act on it. These connected devices improve regular activities and shape smart solutions. However, the vast prospects and benefits brought by IoT lead to the concerns in security. IoT obtains traditional Internet concerns regarding security and also the new ones [1]. The IoT system can be an easy victim to several types of attacks: Physical attacks (Node jamming), Network attacks (Man-in-the-Middle attacks, Denial-of-Service attacks), Software attacks (Trojan horse, Worms, Spyware) and Encryption attacks which is also called as Cryptanalysis attacks [2].

A botnet consists of a number of Internet-connected devices and computers, each of which is running one or more bots. IoT is made up of not only dedicated computers but also household and industrial appliances, vehicles, mechanical sensors, cardiac implant monitors, and other devices which are equipped with IP addresses and can transmit data over a network. The hijacked computers in a botnet are known as zombies or bots. The botnet attack is a dangerous attack amongst the various existing malware present in the network. Botnets can be used to perform Distributed Denial-of-Service (DDoS) attacks, steal data, send spam, and allow the attacker to access the device and its connection. An IoT botnet is a group of hacked computers, smart appliances, and Internet-connected devices that have been co-opted for illicit purposes. The most effective countermeasure against such types of botnet is provided by the Intrusion detection system (IDS) [3].

An IDS is a type of security system that is capable of scanning and monitoring the network traffic for malicious activity and can identify any vicious or abnormal behavior. IDS continuously observe and identify the presence of an active attack by inspecting the vulnerabilities in network traffic. Technically, an IDS is similar to a classification task, i.e., identifying whether the behavior in the network is normal or abnormal. Any classification problem can basically be of two types: binary and multi-class classification. In binary classification, the system generates two types of outputs, i.e, "attack" or "normal". On the other hand, multi-class classification identifies the attack types as well [4]. In this paper, we have made use of binary classification.

With the rapid development of various datasets, DL (deep learning) approaches are mainly considered to mine or extract improved representations out of the given data and can also draw out much more efficient features [4]. The deep learning concept was introduced by G. Hinton et al. in the year 2006 and over the years, it has gone through a spectacular rise in the area of Machine Learning [5]. To build an efficient intrusion detection model, we proposed a unique DL methodology, since DL has the property of automated discovery of abstraction from a raw data set. The methodology we used is to build an IDS for IoT Botnets using LSTM RNN (Long Short-Term Memory Recurrent Neural Network) approach.

his paper presents the LSTM RNN intrusion detection model and its implementation. The model's performance is studied with respect to miscalculation rate, accuracy, precision, true positive rate, and f-1 score. The IDS model is implemented using the TensorFlow, Keras, and Python program language. UNSW-NB15 dataset is used to perform the simulations. This model focuses on detecting only the network attacks, which mainly target the network layer of IoT and is not essential for the attackers to be within the reach of the IoT system to execute such type of attacks [2]. The experimental outcomes exhibit the efficiency of our proposed LSTM RNN model in detecting 5 types of security attacks that an IoT network may encounter. The remaining structure of this paper is as follows. Section II comprises related work within the field of intrusion detection. Section III describes the introduced model for intrusion detection, including the standard UNSW-NB15 dataset, data pre-processing mechanism, and evaluation matrix. Section IV highlights the experimental outcomes and discussions. Section V comprises a result. Finally, Section VI presents the conclusions and the future scope of this research.

II. RELATED WORK

RNN (Recurrent Neural Network) consists of various layers along with feedback loops and is also able to generate past information onward to current time. An RNN comprises loops and those loops permit information to preserve. The hidden layers of RNN models react as information storage like computer memory [6]. RNN constructs a class of strong DNN that uses its internal memory with the help of loops to deal with sequence of data [7]. In these paper the developed model is able to accomplish high accuracy for detecting attacks traffic in those used datasets.The proposed model is able to detect attacks using a reduced UNSW_NB15 dataset, along with more than 95% accuracy with 100% precision.

The author describes the detection of IoT-Botnet at the extension level by logistic regression. The paper describes a maturing model of logistic regression which permits approximately that a device initiating a connection is running a bot. KDD datasets to draw out the features. The extraction of features association is drawn from packet history. Their examination results were high as 98% accuracy. The procedure is based on prototype of logistic regression. it narrates a developed prototype of logistic regression which admits approximating the expectation that a device begins a connection is running a bot [8].

SVM (Support Vector Machine) is a supervised machine learning algorithm that is used for both regression challenges or classification. The SVM categorization is established on the abstraction of decision boundaries. It distinct a set of occurrences having dissimilar class values in the middle of two groups. It bears both binary and multi-class classification. To get results for recognition of patterns from training data, each instance belongs to one or two classes situated on a non-probabilistic binary linear classifier. SVM represents the training instance in space, as the instance of the different classes is split into a clear gap. To get a subset for the high classification accuracy in supervised learning they use feature selection. In this paper, the author had presented an intrusion classification approach was established on a combination of feature selection and SVM classifier using NSL-KDD Cup 99 intrusion detection benchmark datasets. The proposed method supports the classification accuracy of SVM classifier but it uses a reduced set of input features from training data. The author has a beard SVM classifier on some features subsets of NSLKDD Cup 99 dataset and investigational analysis appear that proposed method achieved 91% classification accuracy using only three input features and 99% classification accuracy using 36 input features, while all 41 inputs feature achieved 99% classification accuracy [9].

The solution supply by SVM is theoretically elegant, computationally capable, and extremely effectual in numerous huge practical problems. SVM fabricates a very fine accuracy of 97.32% and has the best execution time of 68.91secs [10].

KNN (K-Nearest Neighbors) is used in Machine Learning for regression and classification problems. KNN algorithms use data and classify new data points situated on similarity measures. KNN is used to carry out the classification considering k-sub-datasets, each of them having alike characteristics putting in Euclidean Distance to figure out the group. IBK is one of most straightforward k-Nearest-Neighbor classifiers [11]. These paper results show that when it comes to the detection accuracy, testing time, and false-positive rate the knn is the most performing classifier among all others [10]. Knn fabricates very advisable accuracy along with the moderate computation time when compared to other classification algorithms. Knn has lowest execution time and an accuracy of 99% [12]. K-NN have an accuracy of 99.44% and the best time of 0.01 sec. K-NN outperformed all other classifiers having the best time[10].

III. PROPOSED MODEL

In this paper a LSTM-RNN model is proposed for the detection of BoT-IoT attack. This model uses the UNSW-NB15 Dataset which contains 9 types of attack. This dataset has been preprocessed and given to our LSTM-RNN model as input.

A. Bot-IoT Dataset Description

KDD'99 dataset has been the common popular and exercised dataset by researchers although it has an inherent issue[13]. It contains a large number of redundant records in the training and test data. [14] suggested that the KDD'99 dataset did not describe the up to-date characteristics for intrusion detection, and it presents the all-inclusive and comprehensive dataset called the UNSWNB15. In KDD'99 dataset have uneven no of attacks in testing and training dataset i.e testing dataset has more 14 types of attack. Their results confirmed that KDD'99 dataset features were less representative that of the features in UNSW-NB15 dataset[15].

The dataset UNSW-NB15 includes 45 features. It is split into training and testing sets containing different Bot-IoT attack types. In [14] the author has studied the UNSWNB15 dataset for the intention of obtaining the most acceptable features and thus introduced a subset with features that increase the efficiency of intrusion detection. This dataset had used tcpdump tool for Capturing network traffic in the form of packets. UNSWNB15 has two subsets of the dataset in the file 'UNSW_NB15_test-set.csv'and 'UNSWNB15_training-set.csv' used for testing and training proposed model. This dataset can be downloaded from :- https://www.kaggle.com/mrwellsdavid/unsw-nb15.

The 'UNSWNB15_training-set.csv' file contains 82,332 records for training and The 'UNSWNB15_testing-set.csv' file 175,341 records for testing. This dataset includes nine types of attack categories namely Backdoor, Worms, Fuzzer, Analysis, Reconnaissance, shellcode, DoS (Denial-of-Service), and Service [16] [17] [18]. In the proposed model all records with these 9 attack categories and normal types are considered as processing the model. Data distribution remains different since it has a separate training and testing sets [19].

The UNSW-NB15 dataset is used by the author in [19], for conducting IoT research because unlike the past benchmark datasets, the UNSW-NB15 datasets show simultaneous attack patterns and modernized normal traffic patterns. The data distribution remains different as UNSWNB15 has separate training-dataset and testing-dataset [19]. KDD'99 and NSL-KDD datasets do not comprehend the nowadays network security concerns and the most advanced attack features. Hence thy are not meeting the modern network security needs[20].

In this research, we have taken the UNSW-NB15 dataset as it covers the latest attack patterns which include normal traffic and nine types of attack traffic patterns. The nine attack categorical classes are explained below:

- Reconnaissance Attack

Reconnaissance is a type of IoT-Bot attack where an attacker engages with targeted system to collect information regarding vulnerabilities like packet sniffing, traffic analysis, and port scanning [20]. The attacker uses port scanning software to detect vulnerable ports.

- Shellcode Attack

A shellcode is a tiny piece of code applied as the payload for the exploitation of a software vulnerability in the system. It is named "shellcode" since the attacker can test the compromised machine by typically starting a command shell and any code that does a similar task is termed as shellcode[21].

- Worms Attack

Worms are malicious software that spreads copies of itself from one system to another and can be effected on the IoT Applications which could harm IoT System [22]. A worm replicates without any human interaction once it enters the system. Mirai and Stuxnet are IoT attacks.

- Generic Attack

The generic attack is to hash a large number of discrete messages till the related hash pops up again with regards to collision resistance[23]. It is sometimes referred to as a birthday attack.

- Exploits Attack

An exploit is a bit of software, i.e chunk of data or vulnerability to generate unintended or a series of commands that take benefit of a bug or unanticipated action to occur on computer software or hardware [24]. Such behavior usually involves things like obtaining control of a computer system.

- DoS (Denial-of-Service) Attack

A DoS attack is a type of attack intended to shut down a system or network, causing it unavailable to its expected users. It floods the target system with traffic or sending it data that triggers a crash of the system. An IoT network can be compromised by DDoS(Distributed DoS) attacks or DoS attacks which make services unavailable to the genuine users due to an overwhelming amount of requests resulting in capacity and resource overload [25].

- Backdoor Attack

The Backdoor Attack is a type of malware attack that is used to acquire an unauthorized introduction to a website by the Attacker. IoT exploding systems such as RTOS and Contik might contain a backdoor where it is probable to reprogram them for making access to private data stored or delivered on the IoT networks [22].

- Analysis Attack

At IoT system networks, these kinds of attacks are targeted. In this attacker first gains relevant network information using port scanning or packet sniffing and later begins attacks on the targeted system [25].

- Fuzzers Attack

In cybersecurity, the fuzzing is the normally automated means of obtaining hackable software defects by randomly serving various permutations of data into a target application until one of those permutations exposes a vulnerability [26].

The UNSW-NB15 dataset has training and testing datasets that contain all these nine types of attacks. The number of samples that contain these attacks is shown in Fig. 1 and Fig. 2.

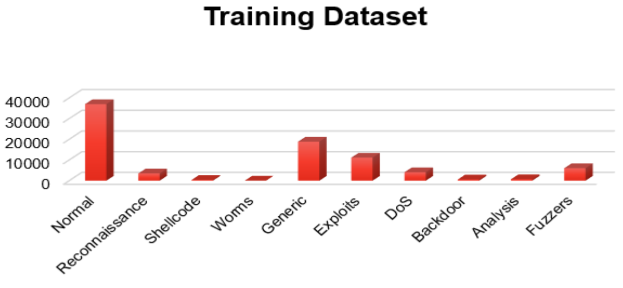


Fig. 1. Training Dataset

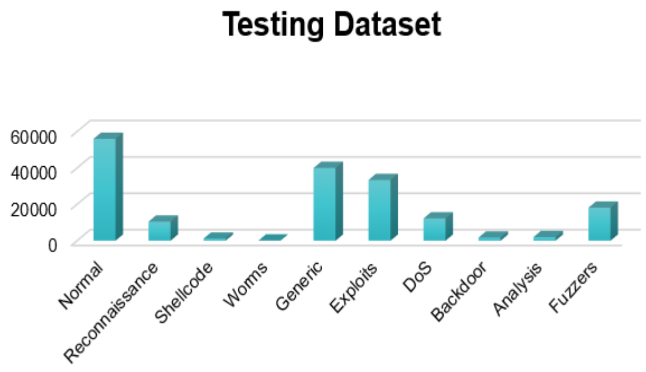


Fig. 2. Testing Dataset

B. Pre-processing of Data

Data pre-processing comprise of normalization, transformation, cleaning, Instance selection, feature extraction and selection, etc. The outcome of data pre-processing is the final training set. Data pre-processing may influence the way in which result of the final data processing can be clarified.

The pre-process method takes the dataset (as a .csv file) as input parameter and rebuild the data samples into a TensorFlow neural network compatible structure which can be used for the purpose of training and testing. The Data class deals with the data preprocessing mechanism and comprise of different preprocessing steps. The preprocessing steps is carried out twice: for the first time, it is called in the IDS (intrusion detection system) for processing the training-set and for the second time it use for processing the test-set. In the first instance, the parameter passed to the preprocessing steps with "UNSW_NB15_training-set_5000.csv" file which consists of training dataset, and in the other instance with "UNSW_NB15_testing-set.csv" file, which has the testing dataset. The preprocess steps summarizes 4 sub-functions to execute the data pre-processing task. The prepared files goes back to main IDS classes and those are used for the purpose of testing and training phases, respectively.

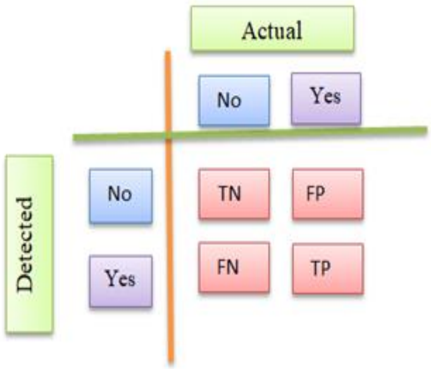
The training dataset is physically operated with the support of approach of the author [27]. where, the author followed the approach of manually putting some non-typical samples in the dataset in sequence to produce the dataset suitable for their research purpose. The welfare of operating the approach is for the input dataset which will be competent for intrusion detection, that will fit the aim of the research. In addition, the approach helps in dealing with the issue of procuring labeled intrusion detection IoT datasets at a high cost.

At this moment, we have followed the approach of manual manipulation and has to extract the features and attack category manually. Thus, our resulting dataset consists of 14 features and two class labels: "Attack" and "Normal".

C. Evaluation Matrix

The confusion matrix is a 2-dimensional matrix which represents the correlation between the detected and actual values. To specify the accuracy of the proposed LSTM RNN model the confusion matrix is applied during the process of testing.

As shown in Fig. 3 True Positive (TP) describes the count of exceptional or unusual samples which are accurately detected by the system. True negative (TN) denote the quantity of normal samples which are consider as normal by the system. False Positive (FP) represents the count of normal samples that are admitted as anomalies. False Negative (FN) refers to the amount of attack samples which are classified as normal.



		Actual	
		No	Yes
Detected	No	TN	FP
	Yes	FN	TP

Fig. 3. Confusion Matrix

The accuracy of the model can be determined by adding TN with TP and dividing by the total number of samples available in the input dataset.

Precision is determined by the ratio of TP to the sum of TP with FP.

The recall is the ratio of TP to the sum of TP and TN.

F1-score is calculated by the ratio of twice the multiplication of recall with precision and sum of precision with recall.

IV. IMPLEMENTATION

Implementation of the model is done in Jupyter Notebook using the library of Tensorflow. The entire implementation process is classified into three phases: data preprocessing, training and testing the LSTM model and evaluation.

A. Data Preprocessing phase

The first stage of implementation is formed by Data Preprocessing. In this stage, the entire training dataset is read and is stored in the memory of the computer. Subsequently, feature extraction is applied. Since Python does not support heterogeneous data types, the non-numeric data items(also known as categorical data) are then changed to numeric values. Dependent variables are encrypted and are followed by data normalization. We build a TensorFlow data structure for storing labels and features, this is done in order to process the features. Since we have employed RNN, the data needs to be reshaped to the respective time steps. In data pre-processing phase the last step is formed by Reshaping.

B. Training phase

The second stage of implementation is the Training phase. In the first step, we have used Keras library to build the LSTM RNN model. The model is compiled and is then followed by model training. It is in this step where the UNSW_NB15_training-set.csv (training-set) file divided into two subsets i.e., the Validation set and Training set, with a 0.33% split ratio (i.e. 33% of the UNSW_NB15_training-set.csv will be used for validation while 67% for training). The Validation subset is being used to analyze the performance of the model after each epoch. After the model is trained, we evaluate the model's performance and redo the training after adjusting the model's parameters until we attain satisfactory performance.

C. *Testing phase*

In this stage of our system, the test datasets are loaded and feed into our trained model for the purpose of testing. The evaluation matrix is then recorded for analyzing our system.

V. RESULTS

In this model, we have used the TensorFlow library Keras deep learning framework to simulate our proposed LSTM RNN model. This model classifies input data samples as Normal or Attack depending on the 14 features which are been selected. The evaluation metrics described in the previous part, shows accuracy, precision, error rate, false-positive rate, recall, true positive rate, and F-1 score which are used to judge the model performance in identifying intrusions.

The model was trained with a sum of 82332 samples. The training samples were gathered from the UNSW_NB15__training-set.csv file. The model was later tested with 175341 test samples. These test samples were obtained from the UNSW_NB15__testing-set.csv file. The proposed model is capable to detect attacks using the UNSW_NB15 dataset, with more than 99% accuracy and with 100% precision. The model produces zero false alarm rates and a pretty low wrong detection rate of 0.05% with an effective recall and f1-score value of 99.8% as shown in Table I.

TABLE I. REPORTED ACCURACY, PRECISION, RECALL, AND F1-SCORE OF PROPOSED LSTM-RNN MODEL

Performance Measure	**Percentage**
Accuracy	0.99998
Precision	1
Recall	0.998
f1-score	0.999

This RNN model gives a very high accuracy compared to Logistic Regression, Supperwise Vector Machine, and K-Nearest Neighbor which is shown in Fig. 4.

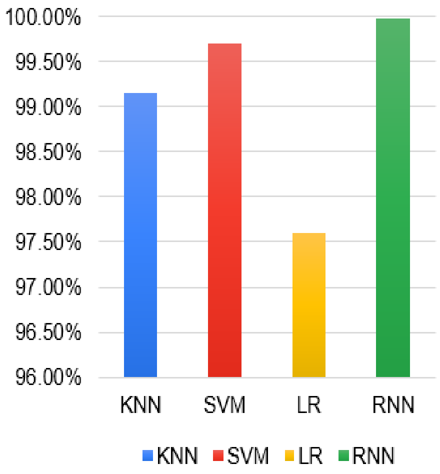


Fig. 4. Accuracy of RNN model vs KNN, SVM, and LR

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a modern IDS model which is based on LSTM RNN for intrusion detection. The LSTM RNN is able to learn the dataset features in detail in the training phase and also performs effective deep learning. This ability is essential in understanding the characteristics of network traffic involved in intrusions to differentiate the abnormal traffic from normal traffic.

We have used Keras deep learning framework and TensorFlow library for the implementation of the proposed new model. The implemented LSTM was applied to the reduced dataset of the UNSW-NB15 dataset, which was used in various published works on IDS in the IoT networks. The detection method was based on binary classification, hence identifying the threat and normal patterns. The model developed achieved higher accuracy than SVM(Support Vector Machine), LR(Linear Regression) and KNN (K-Nearest Neighbors) in recognizing the attack traffic in the dataset used.

For future advancement, more experiments will be carried out to analyze further the proposed LSTM RNN model using vast datasets from the published datasets, mainly the datasets which contain dedicated IoT traffics. Also, the model will be upgraded to increase further its detection accuracy and trade-offs between detection parameters. In addition, the developed model can further be extended to work with real-time data using Wireshark.

REFERENCES

[1] Elkhodr, Mahmoud, Seyed Shahrestani, and Hon Cheung. "The internet of things: new interoperability, management and security challenges." *arXiv preprint arXiv:1604.04824* (2016).

[2] Cui, Zhiyong, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. "Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Forecasting Network-wide Traffic State with Missing Values." *arXiv preprint arXiv:2005.11627* (2020).

[3] Koli, Manoj S., and Manik K. Chavan. "An advanced method for detection of botnet traffic using intrusion detection system." In *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 481-485. IEEE, 2017.

[4] Roy, Bipraneel, and Hon Cheung. "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network." In *2018 28th*

International Telecommunication Networks and Applications Conference (ITNAC), pp. 1-6. IEEE, 2018.

[5] Sun, Yu, Yuan Liu, Guan Wang, and Haiyan Zhang. "Deep learning for plant identification in natural environment." *Computational intelligence and neuroscience* 2017 (2017).

[6] Mike Schuster and Kuldip K Paliwal, “Bidirectional recurrent neural networks,” Signal Processing, IEEE Transactions, vol. 45, no. 11, pp. 2673–2681, 1997.

[7] Z. Cui, S. Member, R. Ke, S. Member, and Y. Wang, “Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction,” pp. 1–12, 2018.

[8] Prokofiev, Anton O., Yulia S. Smirnova, and Vasiliy A. Surov. "A method to detect Internet of Things botnets." In 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 105-108. IEEE, 2018.

[9] Pervez, Muhammad Shakil, and Dewan Md Farid. "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs." In *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*, pp. 1-6. IEEE, 2014.

[10] Suleiman, Mohammed, and Biju Issac. "Performance comparison of intrusion detection machine learning classifiers on benchmark and new datasets." (2018).

[11] C. So-In, N. Mongkolchai, P. Aimtongkham, K. Wijitsopon, and K. Rujirakul,” An evaluation of data mining classification models for network intrusion detection,” In Digital Information and Communication Technology and its Applications (DICTAP), 2014 Fourth International Conference on, IEEE, 2014. pp. 90-94.

[12] N. Rani and R. Kr. Purwar,” Performance analysis of various classifiers using benchmark datasets in weka tools,”International Journal of Engineering Trends and Technology (IJETT), 47(5), pp. 290–294.

[13] Javaid, Ahmad, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. "A deep learning approach for network intrusion detection system." In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 21-26. 2016.

[14] Janarthanan, Tharmini, and Shahrzad Zargari. "Feature selection in UNSW-NB15 and KDDCUP'99 datasets." In *2017 IEEE 26th international symposium on industrial electronics (ISIE)*, pp. 1881-1886. IEEE, 2017.

[15] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." In *2015 military communications and information systems conference (MilCIS)*, pp. 1-6. IEEE, 2015.

[16] Li, Guoquan, Zheng Yan, Yulong Fu, and Hanlu Chen. "Data fusion for network intrusion detection: a review." *Security and Communication Networks* 2018 (2018).

[17] Abdul-Ghani, Hezam Akram, Dimitri Konstantas, and Mohammed Mahyoub. "A comprehensive IoT attacks survey based on a building-blocked reference model." *IJACSA) International Journal of Advanced Computer Science and Applications* 9, no. 3 (2018): 355-373.

[18] Abomhara, Mohamed. "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks." *Journal of Cyber Security and Mobility* 4, no. 1 (2015): 65-88.

[19] Tama, Bayu Adhi, and Kyung-Hyune Rhee. "Attack classification analysis of IoT network via deep learning approach." *Res. Briefs Inf. Commun. Technol. Evol.(ReBICTE)* 3 (2017): 1-9.

[20] Li, Guoquan, Zheng Yan, Yulong Fu, and Hanlu Chen. "Data Fusion for Network Intrusion Detection." (2018).

[21] Al-Duwairi, Basheer, Wafaa Al-Kahla, Mhd Ammar AlRefai, Yazid Abdelqader, Abdullah Rawash, and Rana Fahmawi. "SIEM-based detection and mitigation of IoT-botnet DDoS attacks." *International Journal of Electrical & Computer Engineering (2088-8708)* 10 (2020).

[22] Abdul-Ghani, Hezam Akram, Dimitri Konstantas, and Mohammed Mahyoub. "A comprehensive IoT attacks survey based on a building-blocked reference model." *IJACSA) International Journal of Advanced Computer Science and Applications* 9, no. 3 (2018): 355-373.

[23] Meidan, Yair, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. "N-baiot—network-based detection of iot botnet attacks using deep autoencoders." *IEEE Pervasive Computing* 17, no. 3 (2018): 12-22.

[24] Garip, Mevlut Turker, Peter Reiher, and Mario Gerla. "Riot: A rapid exploit delivery mechanism against iot devices using vehicular botnets." In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1-6. IEEE, 2019.

[25] Li, Guoquan, Zheng Yan, Yulong Fu, and Hanlu Chen. "Data Fusion for Network Intrusion Detection." (2018).

[26] Bertino, Elisa, and Nayeem Islam. "Botnets and internet of things security." *Computer* 50, no. 2 (2017): 76-79.

[27] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research”, *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77, 2007.