International Journal of Electrical Engineering and Technology (IJEET)
Volume 11, Issue 10, December 2020, pp. 412-420, Article ID: IJEET_11_10_052
Available online at https://iaeme.com/Home/issue/IJEET?Volume=11&Issue=10
ISSN Print: 0976-6545 and ISSN Online: 0976-6553
DOI: 10.34218/IJEET.11.10.2020.052

BOTNET ATTACK DETECTION IN INTERNET OF THINGS USING OPTIMIZATION TECHNIQUES

Dr. S. Rethinavalli
Assistant Professor of Computer Science, Shrimati Indira Gandhi College, Tiruchirappalli, Tamil Nadu, India

Dr. R. Gopinath
D.Litt. (Business Administration) - Researcher, Madurai Kamaraj University, Madurai, Tamil Nadu, India

ABSTRACT

The corporation is currently operating in a hyper-connected world in which scores of heterogeneous devices are constantly sharing information in a variety of application contexts such as wellness, improved communications, digital companies, and so on. However, in this case, the wider the genuine wide range of devices and connections, the greater the risk of security risks. Network Intrusion Detection Systems (NIDSs) will be the most popular line of defence in communications networks to combat malicious behaviour and preserve important security services. Nonetheless, there is no standard process for evaluating and comparing NIDSs. Almost all of the ideas fail to disclose critical NIDS validation procedures, making comparison difficult, if not impossible. In this research, an optimization-based method for detecting Botnet attacks in IoT environments is proposed. Botnet detection based on the Genetic Algorithm is proposed, with dynamic thresholds depending on the GA.

Keywords: Botnet Attack, Internet of Things, Intrusion Detection, Optimization techniques, Genetic Algorithm.

Cite this Article: S. Rethinavalli and R. Gopinath, Botnet Attack Detection in Internet of Things Using Optimization Techniques, *International Journal of Electrical Engineering and Technology (IJEET)*, 11(10), 2020, pp.412-420.
https://iaeme.com/Home/issue/IJEET?Volume=11&Issue=10

1. INTRODUCTION

1.1 Internet of Things and Botnet

The use of Internet of Things (IoT) devices is becoming more common. Despite their potential to improve numerous application areas, these devices have inadequate security, which can be exploited by criminals to establish large-scale botnets. Smart homes, agriculture, manufacturing, and smart cities are just a few of the application fields that this paradigm now

encompasses. IoT turns ordinary objects and sensors into Internet nodes, allowing them to connect with humans and other machines in order to do their responsibilities. Most IoT gadgets, unlike traditional computers and smartphones, are not designed to provide Internet access as one of their primary duties. Nonetheless, they acquire and send a large amount of data about the surroundings in which they operate, which is frequently security-sensitive. They can also receive remote commands to act in a variety of scenarios, including those that are life-threatening. They are a suitable target for malicious operations because they are not as safe as other computing devices but also get involved with security-sensitive jobs.

Botnets are one of the risks that will profit the most from IoT security flaws, among many others. Botnets are networks made up of nodes infected with malware that transforms them into bots that attack any target in response to commands from a botmaster [1]. IoT is an excellent environment for botnets for two major reasons [2]. First, the lack of security features in IoT devices facilitates malware transmission and installation. Second, the large number of devices expected to be connected in the coming year provides attackers with an unprecedented amount of vulnerable resources to enable major strikes [15]. The denial-of-service attack carried out by the Mirai botnet in 2016, which brought down a major DNS provider in the United States, demonstrated the threat's catastrophic potential. To detect botnets in IoT devices, use IoTDS [3]. The IoTDS is a host-based solution that analyses the host's CPU and memory use, CPU temperature, and a wide range of ongoing operations to classify malicious or legitimate behaviour. The classification is one-classed by a classifier. The IoTDS architecture is divided into two parts: the IoTDS Agent, which is installed when you look at an IoT device, and the IoTDS Management Console, which is placed on a separate server. The Management Console saves time and effort by inducing new behaviour models, allowing IoT devices free to perform this work, which is undoubtedly costly.

Bot-IoT is a new dataset that includes both conventional IoT-related and other network traffic, as well as a variety of attack traffic commonly utilised by botnets. This dataset was developed on a realistic testbed and has now been labelled, with the label features indicating an attack flow, as well as the attack type and subcategory for prospective multiclass classification purposes [16]. Additional characteristics were created to help classifiers trained on this model enhance their prediction skills. Through statistical analysis, a subset of the original dataset was created, consisting of the 10-best features. Finally, four measures were used to compare the dataset's validity: accuracy, precision, recall, and fall-out. The SVM model that was trained from the full-featured dataset had the highest accuracy and recall, while the SVM style of the 10-best feature dataset version had the highest precision and lowest fallout. With further optimization among these models, it is often argued that even better results could be achieved [17]. It plans to use the BoT-IoT dataset to construct a forensic network deep learning model and assess its reliability in the future.

Botnets, whether they represent a major cyber threat or not, are difficult to combat. Botnets develop in size and complexity as the number of potential nodes on the internet of things, social media marketing, and virtual machines grows. Botnet design is decentralised, meaning there is no single point of failure [18]. Malware payloads are also evolving at a rate of roughly 70%. Botmasters are driven by profit in this low-barrier-to-entry profession that requires no technical expertise and has a low start-up cost. Botnets are difficult to detect because malicious and benign traffic is difficult to separate. While botnet identification occurs during or after an attack, mathematical models allow for the testing of assumptions and potential mitigations ahead of time. The botnet models [5] were examined using the product development lifecycle paradigm, which includes stages such as conception, recruitment, interaction, marketing, and attack execution (CRIME). A hierarchical layered hierarchy of Markov Models (HMMs) will be used to classify the different types of attacks [6]. This primary HMM splits into secondary HMMs

for each type of attack to classify the attacks based on how critical the attack results are and how frequently each attack is to occur.

In the last few years, Internet of Things (IoT) [6] devices have become extremely popular. Almost everyone has at least one IoT device in their immediate vicinity. The idea of staying connected to gadgets in order to track and gather data about day-to-day activities, such as changing home temperature, collecting genuine health data, or just employing a surveillance camera, is quite appealing. According to Cisco's latest data, the total number of linked IoT devices would certainly approach 50 billion by 2020. The Internet of Things (IoT) is constantly growing in popularity, and the danger landscape associated with it is changing in lockstep.

A botnet is a collection of computers connected to the Internet that have been compromised and are now controlled remotely by an attacker using malicious software known as bots [6]. Malicious software is usually malware that the attacker uses. Botnets can be identified by observing the bots in a network's behaviour [19]. The behaviour was observed by looking at the network traffic flow. Botnet behaviour may be studied using classification algorithms, which are useful for selecting criteria that separate botnet traffic from benign traffic. To detecting botnet existence in a network [6] by using machine learning methods to recognise botnet patterns in a network.

2. RELATED WORKS

Noha A. Hikal and M. M. Elgayar [6] suggested a weighted anomaly-based intrusion detection system (IDS) based on an ensemble data pre-processing stage that is applied ahead of time. Because the Internet of Things (IoT) and its applications are becoming more prevalent every day. While gaining benefits from this technology, these enormous numbers of non-smart connected cyber-physical devices have various aspects that have led to serious security challenges, such as node mobility, wireless communications, absence of local security features, scalability, and diversity [20]. The authors presented a framework for detecting botnet assaults in IoT networks, which is based on a machine learning anomaly-based IDS that uses an ensemble data pre-processing technique. The suggested framework is analysed and compared for different learners using a typical dataset; it has achieved detection accuracy of 99.7% with detection times of 30–80 seconds.

Parra, Gonzalo De La Torre, et al [7] Deep learning Phishing and Botnet assaults have been presented as a cloud-based distributed architecture. The model consists of two key security components that work together to detect phishing and application layer distributed denial of service (DDoS): (1) a distributed convolutional neural network (DCNN) model embedded as an IoT device micro-security add-on for detecting phishing and application layer distributed denial of service (DDoS); and (2) a cloud-based temporal Long-Short Term Memory (LSTM) network model hosted from the back-end for detecting Botnet attacks and The ability to execute various degrees of detection at the client and back-end server, thereby using the distributed processing capabilities of client IoT devices and computationally resilient servers, could be a key advantage of [7] suggested strategy.

Vinayakumar, R., et al, [8] suggested a botnet detection system based on two-level deep learning for semantically distinguishing botnets from lawful actions in the application layer when it comes to DNS services. In the first level of the system, the similarity measures of DNS requests are computed using siamese networks based on a predetermined threshold for picking the most often DNS information across Ethernet connections. In the framework's second level, a domain generation algorithm (DGA) based on deep learning architectures is proposed for categorising regular and aberrant domain names. Because of its DNS data potential, the framework is highly scalable on a commodity hardware server.

Pour, Morteza Safaei, et al [9] investigated macroscopic, passive empirical data to give light on this rapidly evolving menace. The author's goal is to classify and infer compromised Internet-scale IoT by observing one-way network traffic alone, as well as to discover, report, and comprehensively analyse "in the wild" IoT botnets. The work begins by introducing a novel darknet-specific sanitization that makes a significant contribution to the field of Internet measurements [21]. Following that, the suggested methodology [9] can fingerprint compromised IoT devices exclusively using darknet data by creating a binary classifier based on a CNN and active measurements.

Venkatraman, S., B. Surendiran, and P. Arun Raj Kumar [10] suggested a Nave Bayesian method that will be used to prevent spam e-mails by combining conceptual and semantic similarity. In smart networks, this approach increases the performance of spam e-mail detection methods. When the Trojan is posing as a legitimate mail server, the gadget can detect it. It translates to enhanced zero-day protection, decreased administrative costs, and no backscatter. When looking at smart environments, it employs conceptual and semantic similarity-based spam for content analysis to discover and avoid unsolicited e-mails created by IoT devices. It is capable of assisting dedicated on-premises, hosted cloud, and cloud.

Xia, Hui, et al [11] presented a dynamic botnet propagation model (for example, the IoT-BSI model) to investigate the effects of two social variables on botnet formation (for example, device spread capability and device identification ability). Measures the effects of heterogeneous credibility of multiple information transmission channels, as well as non-Markovian social contagion power, on determining the discriminating capacity of intelligent systems. On the basis of social theory, the ability to identify is separated into two categories: rational identification and irrational identification.

3. PROPOSED OPTIMIZATION BASED BOTNET DETECTION IN IOT

3.1 Genetic Algorithm for Botnet Detection in IoT

An initial population is created via a genetic algorithm. After that, create a selection pool. After that, it randomly selects parents from the population and breeds them to produce new progeny. On the selected individuals, genetic operators such as mutation and crossover are applied. Better characters are expected in new offspring, resulting in a more optimised solution. To truly make it stochastic, some randomness is introduced. Then fitness is calculated to determine each individual's adaptability. The second pool has people who are more physically fit.

Layered HTTP botnet Detection Steps

Step 1: Take a snapshot of the network's raw packets.

Step 2: Calculate Genetic threshold values for every single layer

Step 3: Perform Layered Detection predicated on Genetic Threshold values.

Step 4: Update the lists on the basis of the Detection.

3.1.1 Initialization

For all of your attacks, a random set of manual thresholds is generated to allow for all conceivable solutions.

3.1.2 Selection

During each consecutive selection phase, the existing population (p rows) is copied from the original pool. The existing population of two parent rows is chosen at random. A rank fitness function is used to choose candidates. Given the p+1th row, the row with the greatest rank is set. In this case, the input should be p rows of 2p output.

3.1.3 Cross Over

Crossover and mutation are genetic operators that contribute to a new generation with improved fitness. Copy the first n from the population selection. For each new population, a couple of parents is chosen from the selection pool to breed, and the new offspring inherits both parents' personalities. Crossover continues to the population, which now has a size of 2p. The output and input both have 2p rows.

3.1.4 Mutation

Mutation is a genetic operator that preserves variety by modifying solutions throughout generations. Copy the first n from the previous solution and generate the second population by altering the feature regarding the randomly picked row for each iteration. There should be 2p rows in both the output and input.

3.1.5 Best N Selection

It is worth noting that mutations will eventually result in a more refined cure. Calculating the ranks of 2p rows from the mutation pool, then sorting, yields the best answer with size p. The variety of highest-ranking rank rows leads to a next-generation solution (p size).

$$\text{Rank/Weight} = \frac{(\text{Success} - \text{Failure})}{\text{TotalNumber}}$$

3.2 Detection Module

3.2.1 DDoS Layer

The detection module is responsible for detecting attacks, particularly DDoS attacks. In DoS attacks, the targeted system is inundated with a tremendous number of requests that it cannot handle. As a result, the system's performance is slowed. As a result, traffic-level characteristics such as source address, destination address, mac address, internet protocol address traffic rate, and packet-level features such as packet contents and errors while inspecting packets are taken into account. The system keeps track of the DDoS layer's genetic threshold value, which is called Dynamic Genetic Threshold Value (DGTV). This layer detects DDoS attacks and updates the database if the wide range of packets exceeds the threshold.

3.2.2 Probe Layer

In probe assaults, the attacker scans the network for computer information in order to detect weaknesses. These network probes may be useful to an attacker planning a future attack. A Probe layer genetic threshold value (PGTV) is maintained by the system. If the number of packets is more than the threshold, this layer records DDoS attacks, and the database is updated.

3.2.3 Root to Local layer

The attacks that are r2L are the most difficult to detect since they involve both network and host level elements. As a result, for identifying r2L attacks, both network level characteristics such as "length of connection" and "service requested" as well as host level features such as "number of failed login attempts" are taken into account. Because the attackers would not have a free account on the victim's machine, they would be desperate to acquire access. Password guessing, for example. A Probe layer genetic Threshold value is maintained by the system (RGTV). This layer detects DDoS attacks and updates the database if the wide range of packets exceeds the threshold.

3.2.4 User to Root layer

The semantic details involved in U2R attacks can be extremely difficult to retrieve at an early stage. Content-based assaults that target an application are common. As a result, it chose information like "number of file creations" and "number of shell prompts invoked" for U2R assaults, while ignoring features like "source" and "protocol" bytes. An attacker with local access to the victim's computer tries to gain superuser rights. The system keeps track of a Probe layer genetic value that is below the UGTV threshold. This layer reports DDoS attacks, and the database is updated if a large number of packets exceeds the threshold.

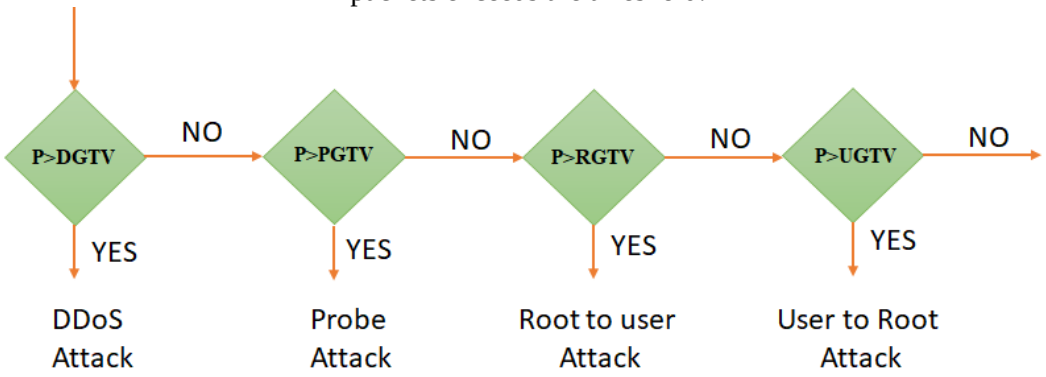

Figure 1 Attack Detection Process

4. RESULT AND DISCUSSION

The KDD cup dataset is used in this research work to evaluate the proposed algorithm in the detection of Botnet attack in IoT. The proposed algorithm is compared with Machine Learning algorithms like Artificial Neural Network (ANN), Support Vector Machine (SVM) and Random Forest (RF). The performance metrics like Precision, Recall, F1 Score and Area Under Curve (AUC) are the used in the evaluation. Table 1 gives the performance analysis of the Botnet attack detection using proposed algorithm, ANN. SVM and RF. Figure 2 depicts the graphical representation of the precision obtained by the proposed Optimization method, ANN, SVM and RF. Figure 3 gives the graphical representation of the recall obtained by the proposed Optimization method, ANN, SVM and RF. Figure 4 depicts the graphical representation of the F1 Score obtained by the proposed Optimization method, ANN, SVM and RF. Figure 5 depicts the graphical representation of the Area Under the Curve (AUC) obtained by the proposed Optimization method, ANN, SVM and RF. From the table 1, figure 2, 3, 4, and 5, it is clear that the proposed optimization based detection method gives better precision, recall, F1 score and AUC than the classification techniques like ANN, SVM and RF.

Table 1 Performance analysis of the Proposed method and existing classification methods in Botnet Attack detection

Performance Metrics	Classification Techniques			
	Proposed Method	**ANN**	**SVM**	**RF**
Precision	**0.974**	0.848	0.828	0.862
Recall	**0.957**	0.837	0.841	0.869
F1 Score	**0.9624**	0.8424	0.8344	0.8654
AUC	**0.972**	0.781	0.773	0.846

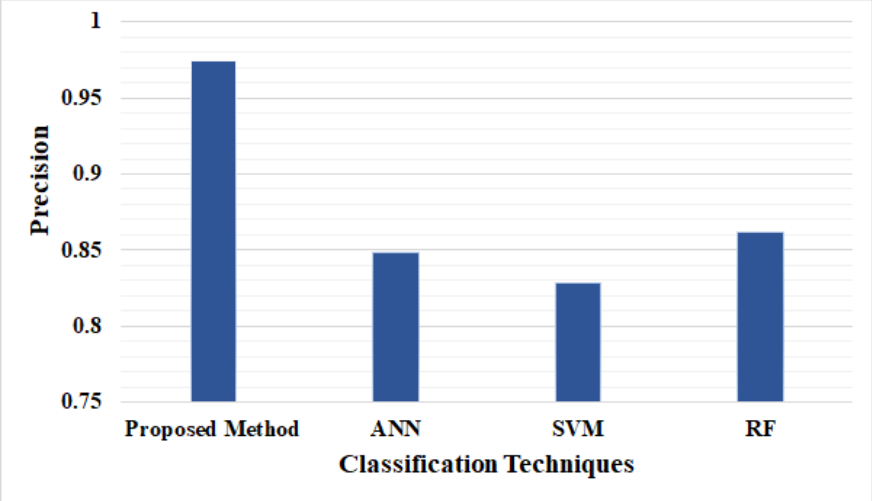

Figure 2 Precision of the Proposed Optimization method, ANN, SVM and RF classification technique

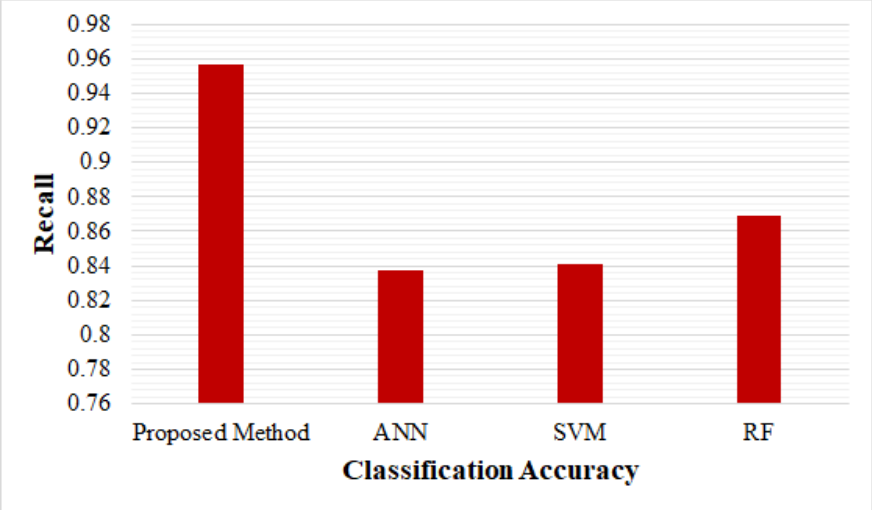

Figure 3 Recall obtained by the proposed optimization method, ANN, SVM and RF classification technique

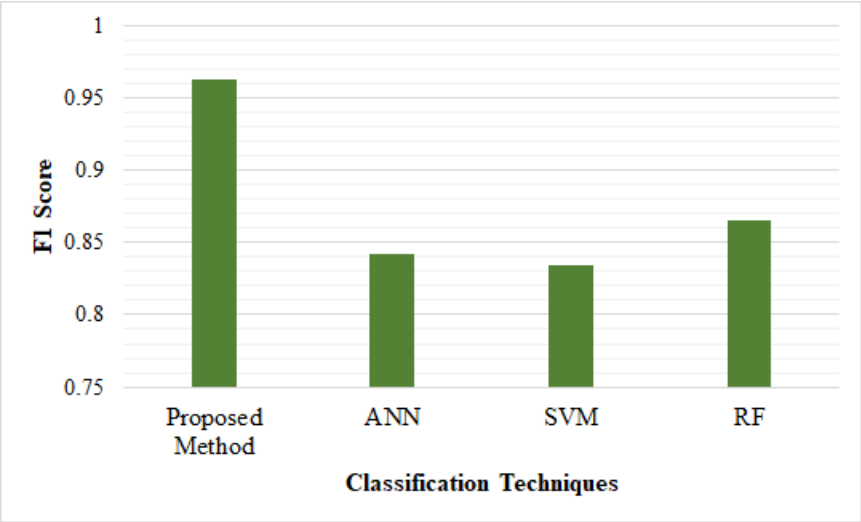

Figure 4 F1 Score obtained by the proposed optimization method, ANN, SVM and RF classification technique

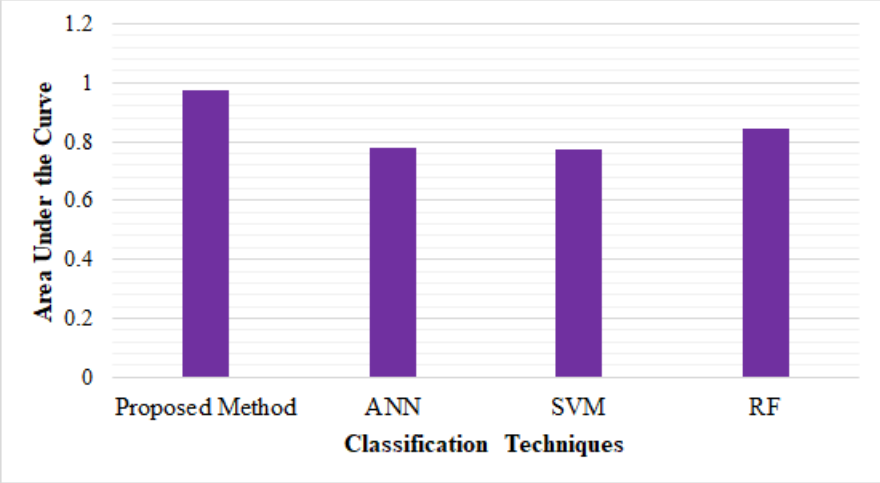

Figure 5 Area Under the Curve obtained by the proposed optimization method, ANN, SVM and RF classification technique

5. CONCLUSION

Almost all of the state-of-the-art approaches try not to appropriately describe or avoid necessary steps followed when you look at the methodology used to gauge their proposals, rendering it difficult to perform a fair comparison evaluation of ML-based NIDSs, also to be confident in regards to the results published by different authors addressing similar forms of problems. The framework suitability happens to be tested with classical ML algorithms and an updated and real network dataset. From the result obtained, it is clear that the proposed method for Botnet attack detection in IoT environment gives better precision, recall, detection accuracy and F1 Score than the existing classification algorithms like ANN, SVM and RF.

REFERENCES

[1] Yu, S.;Wang, G.; Liu, X.; Niu, J. Security and Privacy in the Age of the Smart Internet of Things: An Overview from a Networking Perspective. IEEE Commun. Mag. 2018, 56, 14–18.

[2] Angrishi, K. Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets. arXiv 2017, 1–17, arXiv:1702.03681.

[3] Bezerra, Vitor Hugo, et al. "IoTDS: A One-Class Classification Approach to Detect Botnets in Internet of Things Devices." Sensors 19.14 (2019): 3188.

[4] Koroniotis, Nickolaos, 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 100 (2019): 779-796.

[5] Wainwright, Polly, and Houssain Kettani. "An Analysis of Botnet Models." Proceedings of the 2019 3rd International Conference on Compute and Data Analysis. 2019.

[6] Alshammari, Ahmad, and Mohamed A. Zohdy. "Internet of things attacks detection and classification using tiered hidden Markov model." Proceedings of the 2019 8th International Conference on Software and Computer Applications. 2019.

[7] Banerjee, Mahesh, and S. D. Samantaray. "Network Traffic Analysis Based IoT Botnet Detection Using Honeynet Data Applying Classification Techniques." International Journal of Computer Science and Information Security (IJCSIS) 17.8 (2019).

[8] Angrishi, Kishore. "Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets." arXiv preprint arXiv:1702.03681 (2017).

[9] Hikal, Noha A., and M. M. Elgayar. "Enhancing IoT Botnets Attack Detection Using Machine Learning-IDS and Ensemble Data Preprocessing Technique." Internet of Things—Applications and Future. Springer, Singapore, 2020. 89-102.

[10] Parra, Gonzalo De La Torre, et al. "Detecting Internet of Things attacks using distributed deep learning." Journal of Network and Computer Applications (2020): 102662.

[11] Vinayakumar, R., et al. "A Visualized Botnet Detection System based Deep Learning for the Internet of Things Networks of Smart Cities." IEEE Transactions on Industry Applications (2020).

[12] Pour, Morteza Safaei, et al. "On data-driven curation, learning, and analysis for inferring evolving internet-of-Things (IoT) botnets in the wild." Computers & Security 91 (2020): 101707.

[13] Venkatraman, S., B. Surendiran, and P. Arun Raj Kumar. "Spam e-mail classification for the internet of things environment using semantic similarity approach." The Journal of Supercomputing 76.2 (2020): 756-776.

[14] Xia, Hui, et al. "Modeling and analysis botnet propagation in social Internet of Things." IEEE Internet of Things Journal (2020).

[15] Subhashini, M., & Gopinath, R. (2020). Mapreduce Methodology for Elliptical Curve Discrete Logarithmic Problems – Securing Telecom Networks, International Journal of Electrical Engineering and Technology, 11(9), 261-273.

[16] Upendran, V., & Gopinath, R. (2020). Feature Selection Based on Multi criteria Decision Making for Intrusion Detection System. International Journal of Electrical Engineering and Technology, 11(5), 217-226.

[17] Upendran, V., & Gopinath, R. (2020). Optimization Based Classification Technique for Intrusion Detection System. International Journal of Advanced Research in Engineering and Technology, 11(9), 1255-1262.

[18] Kalaiarasi, K., & Gopinath, R. (2020). Fuzzy Inventory EOQ Optimization Mathematical Model, International Journal of Electrical Engineering and Technology, 11(8), 169-174.

[19] Kalaiarasi, K., & Gopinath, R. (2020). Stochastic Lead Time Reduction for Replenishment Python-Based Fuzzy Inventory Order EOQ Model with Machine Learning Support, International Journal of Advanced Research in Engineering and Technology, 11(10), 1982-1991.

[20] Shanmugavadivu, S. A., & Gopinath, R. (2020). On the Non homogeneous Ternary Five Degrees Equation with three unknowns x^2-xy+y^2=52z^5, International Journal of Advanced Research in Engineering and Technology, 11(10), 1992-1996.

[21] Shanmugavadivu, S. A., & Gopinath, R. (2020). On the Homogeneous Five Degree Equation with five unknowns 〖2(x〗^5-y^5)+2xy(x^3-y^3)=〖37(x+y)(z〗^2-w^2)P^2, International Journal of Advanced Research in Engineering and Technology, 11(11), 2399-2404.