IoT DDOS ATTACK DETECTION USING MACHINE LEARNING

Mahdi Hassan Aysa
Department of Information Technology, Altinbas University
Mahdi3476634@gmail.com

Abdullahi Abdu Ibrahim
Department of Electrical And Computer Engineering, Altinbas University
abdullahi.ibrahim@altinbas.edu.tr

Alaa Hamid Mohammed
Department of Computer Engineering, Karabuk University, Turkey
Aallaaha12@gmail.com

Abstract— **The distribution strategy of a botnet mainly directs its configuration, installing a support of bots for coming exploitation. In this article, we utilize the sources of pandemic modeling to IoT networks consisting of WSNs. We build a proposed framework to detect and abnormal defense activities. According to the impact of IoT-specific features like insufficient processing power, power limitations, and node density on the formation of a botnet, there are significant challenges. We use standard datasets for active two famous attacks, such as Mirai. We also used many machine learning and data mining algorithms such as LSVM, Neural Network, and Decision tree to detect abnormal activities such as DDOS features. In the experimental results, we found that the merge between random forest and decision tree achieved high accuracy to detect attacks.**

Keywords— **IOT, Machine Learning, DDoS, WSNs**

I. INTRODUCTION

IoT systems are frequently enhancing the innovation of our digital landscape. These systems consist of machines with sensing abilities produced to gather data on the situation, which is then transmitted via sink connections to be treated. This data can then be used to discover consumers, increase productivity, or continuation services. IoT sensor systems are characteristically distinctive to popular systems. Sensor accessories are low powered and usually use batteries as their main source of power. Therefore, power performance is a priority[1]. These power limitations mean that machines have restricted processing abilities, which often results in inadequate protection. These unique characteristics have an influence on the propagation of malware and the development of potential botnet threats.

In 2016-17, the Mirai botnet was capable of obtaining friction and, consequently, clutched public attention with a series of high-profile, large-scale DDoS attacks [2]. Using a moderately simple distribution strategy, Mirai quietly spread to many machines, expected to be about 600,000 at its tip. This produced an enormous supply of bots to draw from, resulting in DDoS attacks with a huge force of 620 Gbps against a security blog and 1.1 Tbps against a French Internet provider both in 2016.

Generation is typically a complicated manner to discover and to recognize[3]. This is because complex vulnerabilities develop over various technologies, various settings or receptions can serve as distribution vectors, and we collectively lack experience with comprehensive implementation of new technologies like IoT or IPv6[4].

We implement the system by building an IoT-defence system, a botnet distribution model through IoT-based WSNs, concentrating on IoT's unique properties that distinguish it from other types of network[5]. This issue is not currently well investigated, and we expect to discuss this with the following contributions:

1. Use standard dataset for two well known DDoS attacks.
2. Implement and use two algorithms to normalize and clean data to extract useful features.
3. Implement a framework for detect and analyzing abnormal activity by using standard machine learning algorithms.

The article is constructed as follows: Section 2 provides a background on IoT-based botnets. Section 3 illustrates the proposed framework and defines the parameters and proposed algorithms. In Section 4, we describe our environment setup and display our results. Section 5 explains our conclusions and makes recommendations.

II. RELATED WORKS

IoT sensor networks comprise of remote sensor hubs, which are little gadgets that can detect the climate and make miniature estimations [6]. The devices create a Wireless Sensor Network (WSN) to collaboratively discover and respond to the environment [6] It additionally speaks with IoT-empowered gadgets, for example, switches, giving admittance to a more extensive framework for information recovery and handling. WSNs comprise of sensor hubs and pool hubs. Group hubs fill in as the information assortment center point and entryway to the WSN [7], [8]. Clients can screen IoT-based WSN legitimately, by means of IP-based LAN or distantly over the Internet, and send orders through pool hubs [7].

Sensor hubs comprise of 4 fundamental parts: the force unit, the sensor, the processor, and the radio [7]. The sensor deals with the remote association while estimating natural factors. The processor composes the assignments and deals with changing over the information into signals for transmission, while the force unit comprises of a hub battery pack [7]. The processor additionally deals with the rest cycles that the hubs use to spare force.

Normally the most energy-expending capacity is information trade; The level of intensity required increments drastically for information to travel farther [7]. In this way, tie thickness and examples of dissemination must be deliberately thought of.

The IoT stack is planned comparatively to a TCP/IP stack, with 5 flat layers that characterize start to finish correspondence from the physical climate (Layer 1) to the

978-1-7281-9090-7/20/$31.00 ©2020 IEEE

application (Layer 5). Likewise incorporates extra "helicopters" [7], Speaks to the cycles to be overseen in each layer. These are (an) energy (for example sharing energy between the elements of a hub), (b) versatility (for example following hubs), and (c) assignments (for example exercises of correspondence, message revelation and observation). The conventions in each layer 3 should address vertical tasks [6].

For Botmaster, these cycles can feature weaknesses. For instance: (a) high energy utilization on influenced hubs can cause hub demise, (b) hub traffic can be utilized to join remote organizations in Sybil design assaults, and (c) task projects can be altered to take data[9]. These tasks ought to likewise be considered as a major aspect of automated upkeep of caught hubs. On the energy level, the botmaster needs to restrict the proficiency levels of the robots to abstain from draining energy as this will hurt dissemination gains. At the assignment level, this may incorporate dropping planned errands or incapacitating administrations. In the interim, on the route level, GPS following on portable hubs can uncover new targets [10].

A few IoT correspondence norms are as of now accessible; In this paper, we center around 6LoWPAN (IPv6 over Low Power WPAN) and RPL (Low Power and Lossy Network Routing Protocol). [10], [11] It is extraordinarily intended for LLNs (low force and misfortune organizations) (for example limited organizations, for example, WSNs dependent on IoT [10]. In view of IEEE 802.15.4 organization standard, it gives IPv6 based steering usefulness [6] To associate sensor organizations to IP organizations. The change to IPv6 is important because of the expanding number of web associated gadgets that need interesting identifiers [8]. 6LoWPAN sudden spikes in demand for low-power sensors and adds an interface layer to guarantee similarity between IP-based steering and IEEE 802.1.5.4 low-layers [6].

Mirai caused inescapable interruption with a progression of enormous scope DDoS assaults in 2016 and 2017. As indicated by [2], 65,000 gadgets were tainted inside 20 hours, and the botnet size arrived at its pinnacle of 600,000 bunches. Mirai utilizes worm-based spread that highlights examines for weak gadgets, and is said to target IoT-empowered cameras, switches, printers[12], and video recorders during the "quick output stage" [13]. The malware sends TCP SYN messages to arbitrary IPv4 addresses on ports 23 and 2323.

III. PROPOSED FRAMEWORK

the proposed framework reveals any strange movement by investigating the traffic and ordering this action as indicated by the ordinary exercises recorded for various iot gadgets. the data used in this study is considered as big data as it is extracted from several iot devices[14]. therefore, there are many challenges that we will encounter with this type of data such as it is very large, variety[3], unstructured, can contain missing data and requires a new high-performance processing. the suggested framework divided into the following main phases as shown in the fig.1 below: (1) dataset preparation, (2) data preprocessing, (3) learning and training over the normal and abnormal dataset, (4) testing and evaluations.

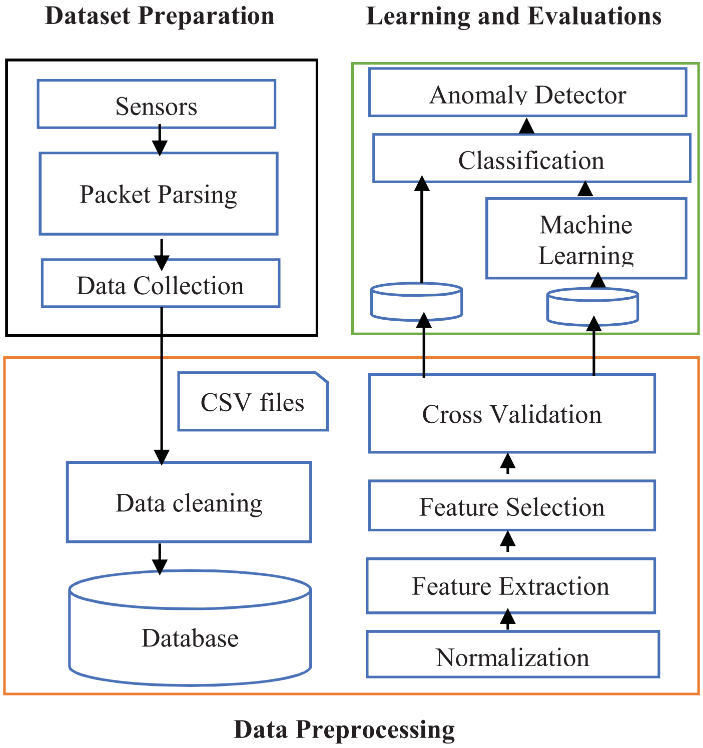


Figure 1: Flow Diagram of Proposed Framework

A. Dataset Preparation

At this phase, the dataset has been collected. It is necessary to collect the data of normal activity first and make sure that they are free of any attacks. This phase included sub-phases such as IoT sensor devices, packet analyzing and data collection. These phases are represented in fig.2.

- IoT devices

In this paper, we used Information from three distinct kinds of Wi-Fi associated IoT camera gadgets. These camera gadgets are: Philips (infant screen), Equipments (security camera) and Simple Home (Safety Camera).

- Packet Capture and Parsing

Many servers are then used to scan vulnerabilities and load attacks. These servers are command & control server (C&C server) and scan server. We analyzed only two of the most common IoT botnet attacks[15]. The primary sort is BASHLITE, a malware assault that taints Linux frameworks by sending virtual information over open telnet. The subsequent kind is Mirai, which chips away at a sweep worker to find and distinguish weaknesses in IoT gadgets that can be assaulted through its IP or Mac address, and afterward introduce malware on those gadgets, confining them from the organization.

- Data Collection

At this stage data are collected before and after the infection of the different devices[14]. Data collected prior to injury are called benign. Attacks were analyzed at different time intervals. The attacks were analyzed at five intervals of time 0.01, 0.1, 1, 3 and 5 seconds respectively. The data was collected prior to the two types of attacks mentioned in addition to the data after the infection in the standard format and structure in the form of CSV in order to overcome the challenge of data varieties.

B. *Data Preprocessing*

At this phase, we prepare large-scale data for learning processes and extracting distinctive patterns[16]. This phase includes many components such as data cleaning, data normalization, feature selection, feature extraction and data splitting into training and testing datasets.

- Data Parsing and Cleaning

Data collected from the previous stage can contain duplicate records, incomplete data or noising data. Therefore, it is necessary to know the different features of the dada. At this stage, the data is divided into a set of different features[17]. Each feature contains many data. Make sure that it is empty and clean and replace any loss on average. It is also necessary to delete and exclude any duplicate data as discussed in algorithm1[18]. Then, data stored in database to find the minimum, maximum, mean and standard deviation of each attribute values.

```
Algorithm 1 - Data Cleaning
Function:
      Input ← (DS) Mixed of normal and abnormal
                  (malicious) Datasets
     Output ← Return cleaned and transformed
Datasets

Data Cleaning:
     array of data ← DS.csv
     for i : array of data. Length
           Remove Redundant data
     End for loop
End of Data Cleaning
```

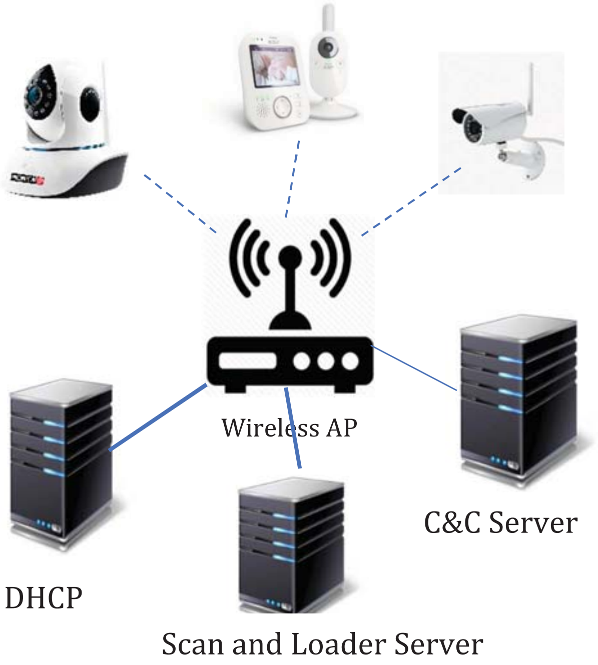


Figure 2: Different Tools used to Collect Dataset

- Data Normalization

Data may contain disparate numbers, different mean and variance which leading to difficulty learning and reducing the efficiency and accuracy of the learning method. Therefore, we used the min-max scaling technique to reduce the negative impact of marginal values as discussed in algorithm2. In this way, all data values in the range from zero to one. This technique usually called feature scaling and its formula represented in eq.1.

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \qquad (1)$$

```
Algorithm 2 - Data Normalization
Function:
      Input ← (DS) Mixed of normal and abnormal
                  (malicious) Datasets
     Output ← Return cleaned and transformed
Datasets
Data Normalization:
     Array of Features ← feature extraction
     for i : Array of Features. Length
          for j: array of data[i]. Length
               min ← minimum value
               max ← maximum value
               new transformed value calculated from
equation 1
          End for loop
     End for loop
End of Data Normalization
```

- Feature Extraction

The information gathered from every client gadget has 115 highlights. These highlights (1) utilize same source have IP address (2) utilizing same source have IP address and MAC address (3) as source have IP and objective host IP (channel) (4) gathered from various traffic. Utilizing various conventions, for example, TCP and UDP (attachments).

- Feature Selection

The motivation behind this stage is to recognize a subset of the key highlights that encourage precise information learning and arrangement. At this stage, it is important to erase all comparative highlights that might possibly be reasonable for other people, and just the fundamental thing highlights ought to be safeguarded. At the point when applied to the information utilized, the properties were diminished from 115 to 40 key highlights. The use of FS has many advantages such as reduces overfitting as minimizing noise and redundancy in attributes, improves modeling accuracy by reducing misleading attributes and reduces training time[19].

Therefore, because of the importance of this step, as we mentioned, there is a way to choose the best and most important features of data. We used the Pearson coefficient technique to evaluate the correlation of the features to understand and measure the level of dependency. Pearson coefficient correlation, according to the Cauchy–Schwarz inequality, has a value between +1 and −1. Positive one means total positive linear correlation, negative one means total negative linear correlation and zero means nonlinear correlation. The basic formula for calculation of Pearson coefficient correlation is shown in eq.2.

$$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y} \qquad (2)$$

where: cov is the covariance, σ_x is the standard deviation of x and σ_y is the standard deviation of y.

Algorithm 3 – Feature Selection

```
Function:
    Input ← (DS) Mixed of normal and abnormal
            (malicious) Datasets
    Output ← Return cleaned and transformed
Datasets
Feature Selection:
    Array of Features ← feature extraction
    for i : Array of Features. Length
        for j: array of data[i]. Length
            x ← feature of i
            y ← feature of i+1
            Cov(x,y) ←calucate covariance between x,y
            $\sigma_x$ ←calucate standard deviation of x
            $\sigma_y$ ←calucate standard deviation of y
            correlation value calculated from equation 2
    End for loop
End for loop
End of Feature Selection
```

- Cross Validation

Cross approval was utilized 10-crease to prepare and test informational collections. 10-overlay cross-approval partitions the informational collection into 10 sections. At that point nine sections are utilized for preparing, and section 10 is utilized for testing. This is then rehashed with another piece as a test piece. Every information section is utilized one time for testing and multiple times for preparing. This is then rehashed multiple times to turn into another test piece. A normal outcome is created without ten runs.

C. *Learning methods*

We use WEKA software to learn on the dataset. WEKA is an open-source software written in Java developed by the University of Waikato in New Zealand. We applied several machine learning algorithms such as Decision Tree (J-48), Linear SVM, Neural Network (Back- propagation), Random Forest.

IV. EXPERIMENTAL RESULTS

A. *Platforms and Tools*

We use different tools to perform our proposed framework. These tools are represented in table1.

Table 1: Tools being used in Preprocessing and Learning Phase

Platforms	Intel core i-7 Windows 10 WEKA 3.8
Server	Apache Server version 2.2.22
Database Engine	MYSQL version 5.5.24
Programming Language	PHP version 5.3.13

B. Dataset Description

We use three standard sets of benign (normal) data and (abnormal or malicious) data collected from three IoT devices. These datasets are downloaded from UCI. Table 2 provides technical details about the properties of datasets being used such as type of features of the analyzed packet, time slots being used, total number of attributes and total number of selected attributes after preprocessing stage and types of attacks being analyzed. Table 3 provides some details of each dataset such as type of IoT device, supported attack and total number of normal records[20].

Table 2: Properties of Data

Number of Devices	3 IoT Sensor Devices
Type of Data	Packet Size
Type of Features	a) Source IP b) Source IP and MAC c) Channel d) Sockets
Time Interval	Five-time intervals a) At 0.01 second b) At 0.1 second c) At 1 second d) At 3 second e) At 5 second
Number of Total Feature	115 Features including each feature * time interval with mean, variance, magnitude, … etc.
Number of Selected Feature	40 Features (4 features * 5 time interval * 2 factors – mean and variance)
Attack	2 attacks (Mirai and BASHLITE)

Table 3: Dataset Description

ID	Device Name	Type of Device	Number of Benign	Supporting Attacks
1	Philips	Baby Monitor	70, 240	Mirai and BASHLITE
2	Provision	Security Camera	62,124	Mirai and BASHLITE
3	Simple Home	Security Camera	19, 528	Mirai and BASHLITE

C. Results and Discussion

On the off chance that the framework says there is an assault and this is valid, at that point that forecast is genuine positive (TP), additionally called exactness or affectability of class positive. In the event that the framework says there is no assault and this is valid, at that point this forecast name is likewise called True Negative (TN) and is additionally alluded to as Negative class explicitness or particularity. In the event that the framework shows an assault and this is a bug, at that point that forecast is a bogus positive (FP). On the off chance that the framework doesn't show an assault however this is bogus, at that point this expectation name is called False Negative (FN). Forecast rates for TP and TN (low FP and FN) are satisfactory and attractive. The precision and review computations are detailed as Equations 3 and 4, individually. Condition, which is a component of F-score, affectability and review. 5 is another regularly utilized presentation metric. MAE has an unmistakable translation as the mean supreme contrast among yi and xi as appeared in Equation 6[21]. After applying four classifiers on three different datasets, we found that decision-tree (J-48), Neural BP network and Random Forest achieved a highest True Positive Rate (TPR) compared with LSVM as represented in table 4 and fig.3. Also, we found that decision-tree (J-48), and Random Forest consumed minimum time compared with Neural BP, as represented in table 4 and fig.5, using the same iterations and threshold.

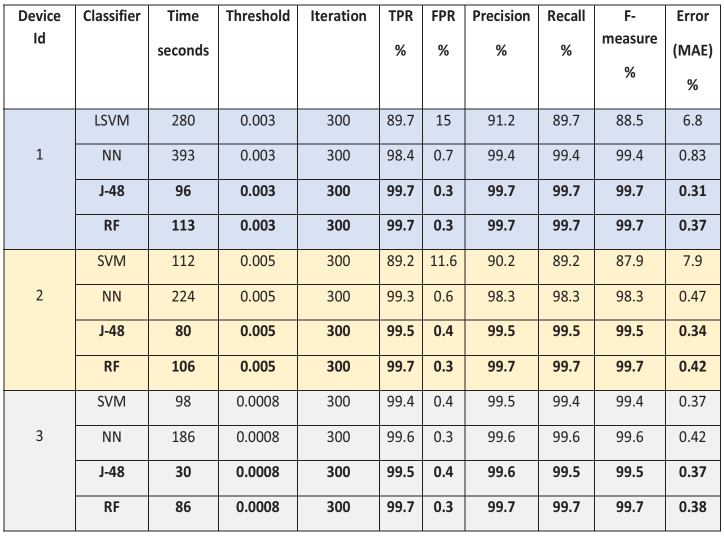
$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$Precision = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{5}$$

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{6}$$

Table 4: Comparison Between Different Classifier



Device Id	Classifier	Time seconds	Threshold	Iteration	TPR %	FPR %	Precision %	Recall %	F-measure %	Error (MAE) %
1	LSVM	280	0.003	300	89.7	15	91.2	89.7	88.5	6.8
	NN	393	0.003	300	98.4	0.7	99.4	99.4	99.4	0.83
	J-48	**96**	**0.003**	**300**	**99.7**	**0.3**	**99.7**	**99.7**	**99.7**	**0.31**
	RF	**113**	**0.003**	**300**	**99.7**	**0.3**	**99.7**	**99.7**	**99.7**	**0.37**
2	SVM	112	0.005	300	89.2	11.6	90.2	89.2	87.9	7.9
	NN	224	0.005	300	99.3	0.6	98.3	98.3	98.3	0.47
	J-48	**80**	**0.005**	**300**	**99.5**	**0.4**	**99.5**	**99.5**	**99.5**	**0.34**
	RF	**106**	**0.005**	**300**	**99.7**	**0.3**	**99.7**	**99.7**	**99.7**	**0.42**
3	SVM	98	0.0008	300	99.4	0.4	99.5	99.4	99.4	0.37
	NN	186	0.0008	300	99.6	0.3	99.6	99.6	99.6	0.42
	J-48	**30**	**0.0008**	**300**	**99.5**	**0.4**	**99.6**	**99.5**	**99.5**	**0.37**
	RF	**86**	**0.0008**	**300**	**99.7**	**0.3**	**99.7**	**99.7**	**99.7**	**0.38**

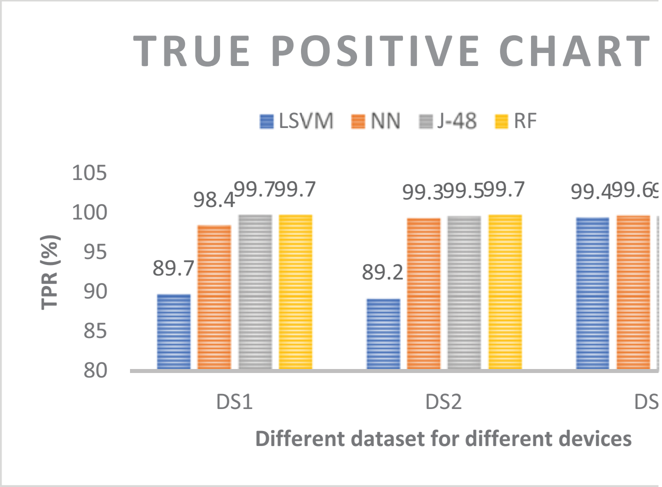


Figure 3: True Positive Chart for each Datasets using Different Classifiers

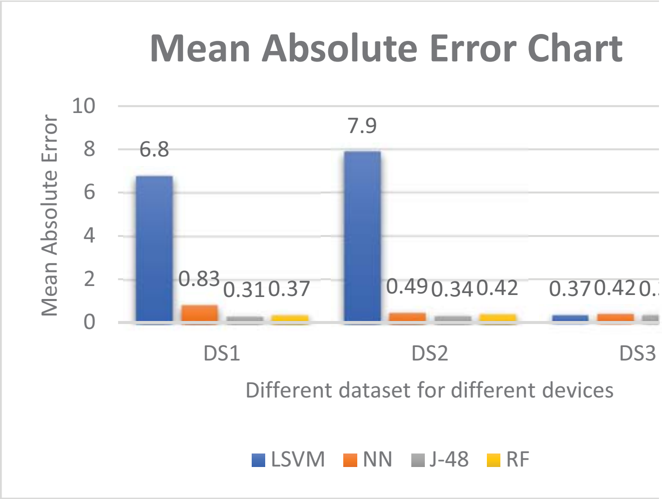


Figure 4: Mean Absolute Error Chart for each Datasets using Different Classifiers

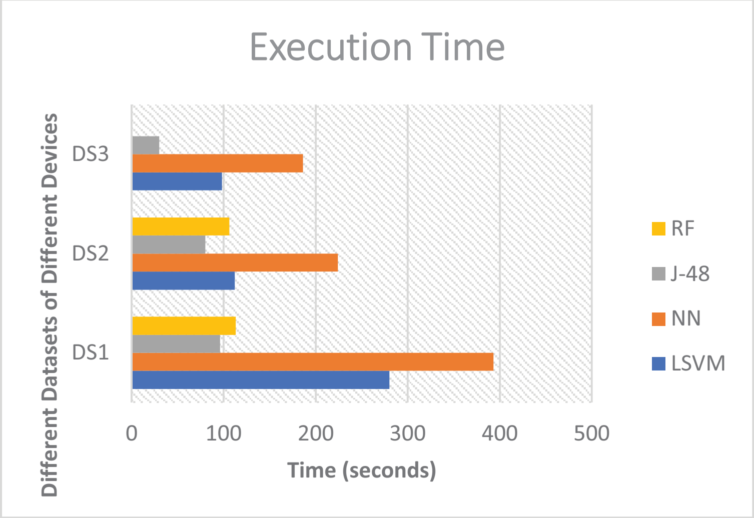


Figure 5: Total Execution Time Chart for each Datasets using Different Classifiers

V. CONCLUSIONS

The distribution strategy of a botnet mainly directs its configuration, installing a support of bots for coming exploitation. In this article, we utilize the sources of pandemic modeling to IoT networks consisting of WSNs. We build a proposed framework to detect and abnormal defense activities. According to the impact of IoT-specific features like insufficient processing power, power limitations, and node density on the formation of a botnet, there are significant challenges. We use standard datasets for active two famous attacks, such as Mirai. We also used many machine learning and data mining algorithms such as LSVM, Neural Network, and Decision tree to detect abnormal activities such as DDOS features. In the experimental results, we found that the merge between random forest and decision tree achieved high accuracy to detect attacks.

REFERENCES

[1] A. Iqbal, M. A. Suryani, R. Saleem, and M. A. Suryani, "Internet of things (IoT): On-going security challenges and risks," *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 11, p. 671, 2016.

[2] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, 2017.

[3] A. G. Mushref, A. H. Mohammed, and O. Bayat, "Rayleigh Leistungs Relation and Rician Fading Channels in Qam Using Simulink Environment," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–5.

[4] S. N. Swamy, D. Jadhav, and N. Kulkarni, "Security threats in the application layer in IOT applications," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017, pp. 477–480.

[5] M. M. Hamdi, L. Audah, S. A. Rashid, A. H. Mohammed, S. Alani, and A. S. Mustafa, "A Review of Applications, Characteristics and Challenges in Vehicular Ad Hoc Networks (VANETs)," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–7.

[6] K. Zhao and L. Ge, "A survey on the internet of things security," in *2013 Ninth international conference on computational intelligence and security*, 2013, pp. 663–667.

[7] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the Internet of Things (IoT): A security taxonomy for IoT," in *2018 17th IEEE International Conference On Trust, Security And*

Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018, pp. 163–168.

[8] A. H. Mohammed, R. M. KHALEEFAH, and I. A. Abdulateef, "A Review Software Defined Networking for Internet of Things," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–8.

[9] S. KURNAZ and A. H. Mohammed, "Secure Pin Authentication in Java Smart Card Using Honey Encryption," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–4.

[10] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: a review," in *2012 international conference on computer science and electronics engineering*, 2012, vol. 3, pp. 648–651.

[11] A. H. Mohammed, R. M. KHALEEFAH, and A. H. AlMarzoogee, "The Method Of Calibration Compensation For Fiber Nonlinearity-A Review," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–8.

[12] Z. A. Sahib, O. N. Uçan, M. A. Talab, M. T. Alnaseeri, A. H. Mohammed, and H. A. Sahib, "Hybrid Method Using EDMS & Gabor for Shape and Texture," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–6.

[13] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Comput. networks*, vol. 31, no. 8, pp. 805–822, 1999.

[14] A. M. Shantaf, S. Kurnaz, and A. H. Mohammed, "Performance Evaluation of Three Mobile Ad-hoc Network Routing Protocols in Different Environments," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–6.

[15] A. H. Mohammed, A. M. Shantaf, and M. Khalaf, "The Probe into Reflection Mobile Radio Propagation," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–4.

[16] A. H. Mohammed, "A REVIEW SOFTWARE DEFINED NETWORKING FOR INTERNET OF THINGS," *2020 Int. Congr. Human-Computer Interact. Optim. Robot. Appl.*, no. August, 2020.

[17] A. H. Mohammed, "An Optimum Design of Square Microstrip Patch Antenna Based on Fuzzy Logic Rules," *2nd Int. Congr. Human-Computer Interact. Optim. Robot. Appl.*, 2020.

[18] T. Jan and A. S. M. Sajeev, "Ensemble of probabilistic learning networks for IoT edge intrusion detection," *Int. J. Comput. Networks Commun. Vol*, vol. 10, 2018.

[19] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Anal.*, vol. 1, no. 1, p. 9, 2016.

[20] A. H. Mohammed, M. M. Hamdi, S. A. Rashid, and A. M. Shantaf, "An Optimum Design of Square Microstrip Patch Antenna Based on Fuzzy Logic Rules," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–7.

[21] M. Khalaf *et al.*, "IoT-Enabled Flood Severity Prediction via Ensemble Machine Learning Models," *IEEE Access*, vol. 8, pp. 70375–70386, 2020.