International Journal of Advance Science and Technology
Vol. 29, No. 10S, (2020), pp. 8006-8016

IoT Malware Detection using Machine Learning Ensemble Algorithms

Santhadevi D[1], Janet B[2]

Abstract

Internet of Things and Network Technology together increases the complexity of cybersecurity. Current cyber-attack detection approaches are ineffective, so that they are easy targets for cybercriminals. A type of malware that forces a genuine danger to the Internet security is known as botnets. This malware misuses a few vulnerabilities of IoT gadgets to infect them and perform huge scale Distributed Denial of Service (DDoS), Internet Relay Chat (IRC) and spam assaults. Cyber threat hunting still remains an open challenge. Machine Learning techniques are more appropriate methods to detect these attacks. In this paper, two ensemble machine learning models like Ada-Boost and Stochastic Gradient Descent Classification (SGDC) are used to improve the accuracy of botnet detection in the Netflow data. The benchmark dataset of CTU -13 is used to train and test the machine learning models. The accuracy of detecting the Internet Relay Chat attack is improved in the SGDC model by 98.31% and F1 score by 97.47%. Ada-Boost achieved a higher percentage than SGDC in DDoS and Spam attacks and the accuracy score achieved by this model is 98.73%. The comparative summary of ensemble classifier models are demonstrated. These ensemble models are performing well for large scale data set and improve the detection strategy of malware greatly.

Keywords— Botnet detection, Ada-Boost, Stochastic Gradient Descent classification, Machine Learning, Malware detection, Internet of Things, Ensemble algorithm.

1. INTRODUCTION

Malware is a malicious software. Intent of creating the malware is to damage the device, steal data and create a mess on the network[1]. Some of the malware are viruses, worms, trojan, spyware, ransomware, and botnet. Virus mostly appears as an executable file. Once the executable file is activated then it starts infecting files on the system. It can be spread uncontrollably and damages systems core functionality. Worms infect the network device by using the network interface. It will affect both the local network device and devices across the internet[2]. It spreads the malware consecutively from one affected device to another through the network. Trojan masks as legitimate software and creates backdoor entry on the system security. This backdoor entry allows other malware to enter into the system. Spyware is hidden software that works in the background to keep a spy on the system. It keeps monitoring user's activity on the system by collecting username and password of the financial transactions, credit and debit card information, surfing habits and more. Ransomware is a hazardous software hidden in the web link or file that is attached with the mail. When that hidden software is activated by the human action it starts threatening the system file with encryption[3]. Botnets are one of the most hazardous network assaults today as they involve the use of the brute and intelligent attacks of huge and coordinated hosts[4]. These big groups of hosts are grouped into so-called zombies or bots. The botnet is controlled by a single command and control (C&C) infrastructure. The attacking host is obstructed by botnets, and the attack host is indirectly divided from the victim utilizing a zombie host layer, which is arbitrary in time segregated from the botnet assembly. Employing massive distributed denial-of-service attacks, Botnets may induce severe network disruptions, and the prospect of such interruption may cost businesses substantial extortion charges[4]. They form a vast majority of internet spam today. The botnet is used to collect delicate data on a flourishing organized crime market, whether private, corporate or governmental. It is definitely a reusable, renewable resource for stealing data[5].

ISSN: 2005-4238 IJAST

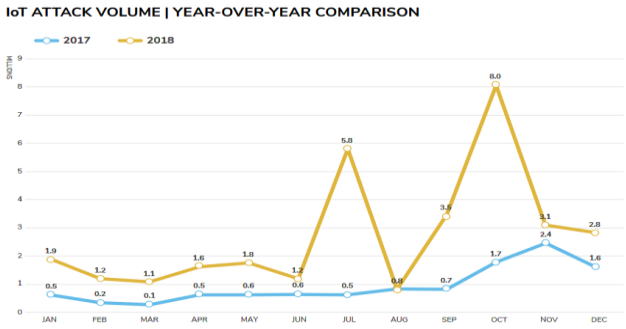


Fig. 1. Attacks tracked by sonic wall – 2019 Cyber Threat report [19].

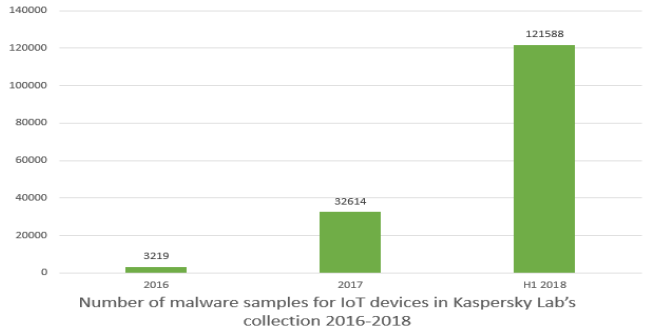


Fig. 2. Kaspersky Lab's report on malware threat on IoT devices from 2016 to 2018 [20].

Fig. 1 and Fig. 2 are malware threat reports generated by some of the most prominent cybersecurity labs of Sonic wall and Kaspersky. The report shows that malware attack on the IoT devices are escalating every year. Attacks are mostly happening based on the compromised devices. The devices are taken control by default credentials and un-updated software or protocol flaws. The command and control server has control over these weak devices and create their own army of bots. These botnets are used to generate a DDoS attack on the network. Early detection of the malware flow in the network is one of the major tasks in the cybersecurity field. The objective of this paper is to detect malware using ensemble machine learning models. Ensemble classification algorithms of Ada-boost and Stochastic Gradient Descent are used to improve the accuracy of detecting the botnet in the network flow communication. The section 2 covers the current study in the area of detecting botnet in the network flow data and various methodology used. Section 3 describes the data set of CTU-13. Section 4 expounds the methods to detect the malware activity in the network flow data. Section 5 discusses the result analysis of different machine learning models. Section 6 covers the conclusion and future work of this paper.

2. LITERATURE REVIEW

The number of interconnected devices is increasing exponentially in the name of the Internet of Things. If any one of the device is compromised, then all interconnected devices are affected by the malware that will grow like botnets. Therefore early detection of such botnet attack is a key in the direction of securing the infrastructure. P. Garcia et. al.[8], explored the advantage and disadvantage of a widely used attack detection model on the network. They have used the machine learning models for detecting a particular type of attack in the network flow. S. Garcia at. el.[6], compared seven different types of botnet attacks that are examined using the machine learning model in the real network data caught across the computer network. The botnet operation has been performed using command and control architecture. The command and control server is used to send the commands to infect vulnerable nodes through the legitimate IRC channel. IRC is Internet Relay Chat channel for network communication [9]. The paper has analyzed the traffic stream from the communication channel, and they have used a statistical approach for anomaly detection to identify the malicious activity correlated to botnet [10]. Stevanovic et. al. have used supervised machine learning models to

classify the network traffic which is usual or abnormal[11]. Livadas et. al. [12], developed Multistage Bayesian network classifier, the first step is to analyze the IRC traffic and second step is to detect the botnet activity in the communication. Random forest model is effectively used in analyzing various network traffic studies [13]. This paper analyses the comparative analysis of ensemble algorithm using machine learning techniques.

3. DATA SET

CTU-13 dataset comprises of 13 different malware captured in the real-time network environment. It is labeled as Botnet and Normal. It has captured the network traffic coming from infected hosts, normal traffic from verified normal hosts and the rest of all network traffic as background traffic. The downloaded file formats are Pcap files, Binetflow files, Biargus files. Pcap file contains the traffic information of all packets with all the header and payload information. Binetflow files have the information of all traffic (ie, Botnet, Normal, Back-ground) and text files with bidirectional flow of network data. Biargus files have all traffic information and binary files with bidirectional network flow data[6].

A. Feature Extraction

- On-premise of the time window extract the total Netflow information corresponding to the IP address
- Source IP address is utilized to bunch each time window
- Statistics determined for each time window of source IP address as tails
    a. Data flow information in the count
    b. Amount of transmitted bytes
    c. The average amount of bytes per Netflow
    d. Average message time with each distinctive IP address
    e. Amount of distinctive endpoint IP address
    f. Amount of distinctive endpoint ports
    g. Often used protocols (ex., TCP, ICMP, UDP, etc.)

The multiscale analysis is used to identify possible infected IP addresses[7]. To create or build a complex model, the multi-scale factor is used for extracting feature vector. It can be achieved by scaling the size of the time window where the feature vectors are calculated. Malware behavior is analyzed over time. Different time scale is used to observe the action of different kinds of malware. For example, spam-sending and scanning can be seen with short time windows and the flow of the significant volume of data. The features of the number of opened connections and number of distinct port numbers are used to identify the spam attack. The features of connection information and entropy IP address are used to determine the DDoS attack.

4. METHODOLOGY

Bots are detected using network monitoring techniques based on passive network monitoring and analysis which is useful for detecting only known attacks by stored patterns. Real time network monitoring and analysis for bot detection is a challenging task. Machine learning is the one of the best technique to detect flow of bots in the network. Rapid encounter of botnet is the most important trial. There are many machine learning algorithms which were used for botnet detection. Ensemble algorithms are more efficient and increases the accuracy in detection of bots in the network. Two such machine learning algorithms are used here to detect the malware attack in the network.

B. Ada-Boost

Boosting algorithms are mostly used in data science to improve the accuracy of the model. These algorithms are made up of a combination of many algorithms, and output is the combination of all combined algorithms. Boosting refers to using the weak learners to build an active learner. To convert weak learner to strong learner, higher weights are assigned for wrong predictions. In each iteration, different weak distribution rules are applied that are combined to form a strong distribution rule at the end [14]. Selection of distribution rule is a crucial phase; it is a three-step process [14].

ISSN: 2005-4238 IJAST

- Step1: Initially, base learners assign equal weight to each distribution and observes equally.
- Step 2: For each wrong prediction, higher value is assigned, and the next distribution algorithm is applied.
- Step 3: Step 2 is repeated to achieve higher accuracy.

This process sooner or later increases the prediction accuracy of the machine learning model. Fig. 3 explains how Ada-boost works. In box1, equal weight is assigned to each sign after applying the first algorithm the decision line only predicts two + signs rest three are not predicted which have assigned higher weights. In box2, next decision algorithm is applied, it predicts only two (-) signs. In box3, the same procedure is used. After applying these three decision lines it can accurately predict the values which are shown in box4.

SGDC algorithm:
Input: Netflow aggregated file X_i features with n records

For j in range (n):

$$Y' = \sum_{i=1}^{n}(X_i * W_i) + b_i$$

$$\theta_j = \theta_{j-1} - \alpha(Y^j - Y'^j)X_i^j$$

End For

α- learning rate
θ_i- parameter value after SGD
X_i – input parameter
W_i- weight of corresponding input
b_i- bias
n- no. of iteration
Y'^j - expected output on j^{th} iteration
Y^j- actual output on j^{th} iteration
X_i^j – input parameter of aggregated pcap file based on time window

To optimize the performance of the algorithm it must tune some parameters, which are

- n_estimators: weak learners are controlled by these estimators.
- Learning_rate: To control the weak learner contribution in the final grouping
- Base_estimators: To select different ML algorithms

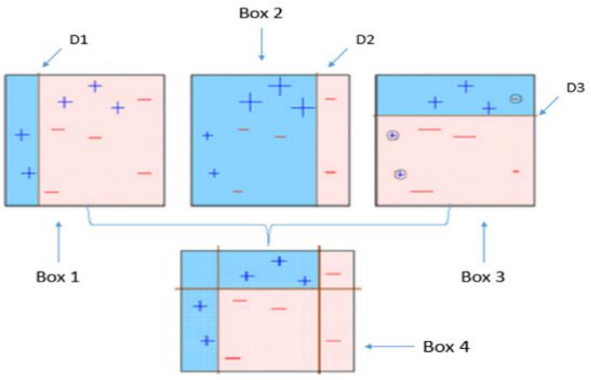


Fig. 3. Example of process of Ada-boost algorithm

C. Stochastic Gradient Descent Classification (SGDC)

Most of the machine learning and deep learning algorithms uses this technique of Gradient Descent (GD) for optimization. It is a popular optimization technique, a mathematical description of GD is a convex function that uses partial derivatives of a set of parameters concerning its inputs. These sets of parameters are initially assigned some values. After some iteration, gradient descent finds the optimal values of the parameters using calculus; finally, it will find the minimum possible value of the given cost function [15]. Stochastic means a system which is linked with a random probability. In SGDC, instead of selecting the whole data set for each iteration, some sample data is randomly selected for

each iteration. In general gradient descent, whole dataset has been taken as the input for each iteration. Even though, passing the whole dataset is good for getting the minima in less noisy and less random way. But, when we use a huge data set the real problem occurs. Hence, for talk about the computation is also very lavish to perform [16]. In SGDC, each iteration sample is randomly shuffled and only a single sample is taken that represents the batch size of one. It will take less time to reach minima. It will take longer iteration but is still computationally less expensive than typical gradient descent.

5. EVALUATION OF RESULT AND DISCUSSION

The data set from [6] was experimented and compared with two different classification algorithms for predicting botnet. In this process, the Netflow folder is aggregated by time window and intensified with predicted labels produced by various methods. The following steps are executed to get the concluding result[7]

- The time window is used to separate the Netflow labels for comparison.
- The output of the Netflow folder is compared with the predicted output. From that true positive (TP), true negative(TN), false positive(FP), false negative(FN) values are calculated.
- For each time window, the following performance indicators are calculated: accuracy, precision, recall, and F1 score.
- The metric TP count is measured based on IP- based performance because the Netflow level for a time window is huge, practically it does not make sense due to administrator having a massive volume of that data.

IP address-based metric measurement:

1. TP: At least once the malicious node IP address is identified as malicious in time window. Then TP count is increased.
2. TN: Non-malicious node IP address is identified through normal in the time window. Then TN count is increased.
3. FP: At least once the malicious node is identified as normal IP address in the time window. Then FP count is increased.
4. FN: Non-botnet IP address is identified as a botnet in the time window. Then FN count is increased.

Time is also incorporated with these metrics to emphasis early detection of malicious IP node. To calculate the time-based component of perfection function (pf), the following formula is used:

$$f(n) = 1 + e^{-\propto n} \quad (1)$$

,

Where n is Time window, α is learning rate which is set as a constant value of 0.01.
The performance metrics are determined as:

$$TP_t = \frac{TP * pf(n)}{N_b} \quad (2)$$

Where, TP_t is time grounded true positive count, perfection function pf and N_b is the number of unique malicious node IP addresses present in the time window.

$$FN_t = \frac{FN * pf(n)}{N_b} \quad (3)$$

For normal Netflow communication, time window comparison is not necessary, so FP_t and TN_t are not subject to the time window, which is defined as follows:

$$FP_t = \frac{FP}{N_{normal}} \quad (4)$$

$$TN_t = \frac{TN}{N_{normal}} \quad (5)$$

True Positive Ratio (TPR):

$$TPR = \frac{TP_t}{FN_t + TP_t} \quad (6)$$

Where N_{ormal} represents the quantity of distinctive normal IP addresses existing in the time window. Succeeding the above notations, final performance metrics are calculated as follows:

i. Accuracy: $A = \frac{TP_t+TN_t}{TP_t+TN_t+FP_t+FN_t}$ (7)

ii. Precision: $P = \frac{TP_t}{TP_t+FP_t}$ (8)

iii. Recall: $R = \frac{TP_t}{TP_t+FN_t}$ (9)

iv. F1-Score (10)

$$\mathrm{F} = 2 * \frac{\mathrm{P} * \mathrm{TPR}}{\mathrm{P} + \mathrm{TPR}}$$

D. Evaluation of Dataset

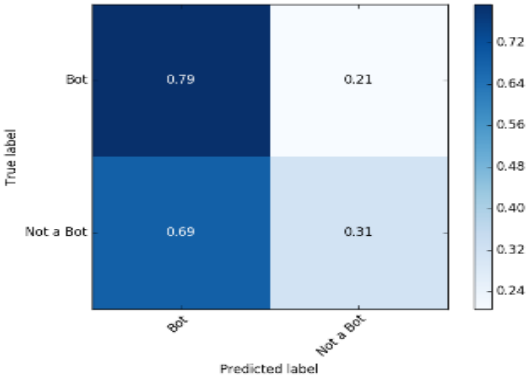
For evaluating the algorithm, we used the dataset of CTU-13[6]. In this data set, several botnet attacks are represented in 13 different scenarios. Each scenario's traffic information is collected in the form of Netflow that has been recorded separately as a CSV file. The file contains the information of starting time of the logged Netflow, interval, code of behaviour (eg., TCP, UDP), source IP address, source port, direction of the recorded communication, destination address, protocol state, source type of service, destination type of service, complete exchange of packet information from source to destination, entire bytes transferred, sum of bytes directed by the source, label (e.g., background, botnet, normal)[18]. The experimental setup used here is GPU system with the configuration of 16 GB RAM, 3.5 GHz processor speed. Python 3 is used for the implementation of Machine Learning algorithms. From Netflow information, we have used 14 attributes, and 15th attribute is the label that is assigned by the classifier.

E. Result Evaluation

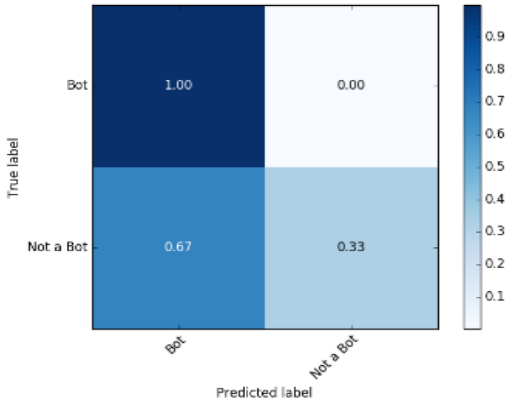
Two different ML algorithms are used that has been analysed in different scenario and configuration. Seven different types of malware in 13 different scenarios are aggregated under two classes which are Not a Bot and Bot[6]. 70% of aggregated data has been taken to train the model. 30% of aggregated data has been taken for testing the model. Higher accuracy rate obtained by this model for predicting the IRC attack is 98.31%, DDoS attack prediction is 97.25% and SPAM attack prediction is 98.39%. Table 1 represents the performance metrics of the Machine Learning algorithm to detect botnet activity in the network that is obtained from 3 scenarios detailed in the testing dataset. Scenario 1 matches to IRC – the type of botnet that sends junk mails which is better predicted by SGDC model when compared with Ada-Boost model. The accuracy of this model achieved is 98.31%. In Scenario 2, C&C servers create an army of the botnet and try to attack the single server. The DDoS attack is generated in this scenario. Ada boost gives a higher accuracy rate of 99.31%. SGDC model also performed well and achieved an accuracy of 99.25%. In Scenario 3, botnet scans the Mail servers for several hours, and it targets the different IP address sent from single source IP. Both algorithms performed well with better prediction rate. Ada- boost algorithm has given accuracy

of 98.73%. Random forest algorithm performs better prediction than the booster algorithm in many scenarios. Fig. 4 represents the confusion matrix of both the algorithms at each attack scenario. In IRC attack, SGDC classified accurately all windows out of around 40,000 windows in testing data. In DDoS attack, SGDC model mis-classified only 7 percent of windows in testing data. Due to imbalanced data set false positive ratio is increased and the accuracy of classifying the IRC attack with Adaboost model is varied. Because of that around 21 percent of windows are mis-classified. In spam and DDoS attack, good percentage of classification rate is achieved with both the models.

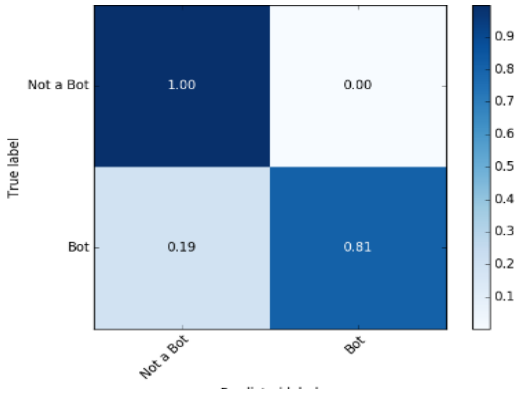
IRC- Adaboost



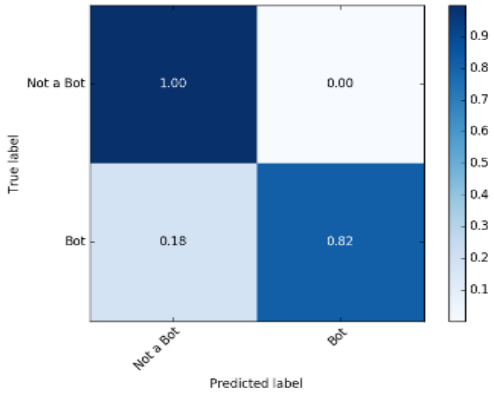
IRC- SGDC



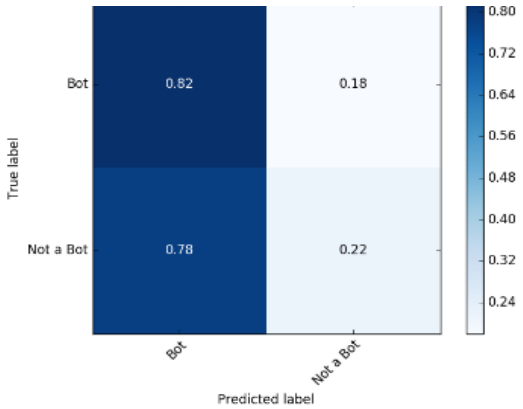
Spam- Adaboost



Spam- SGDC



DDoS- Adaboost



DDoS- SGDC

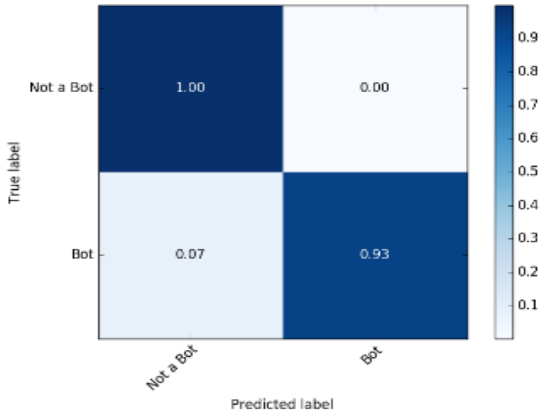


Fig. 4. Confusion Matrix of 2 different classification algorithms on the botnet attack scenario

Table- I: Comparison of classification algorithms performance metrics

IRC Attack				
Algorithm	***Accuracy***	***Precision***	***Recall***	***F1 Score***
SGDC	**0.9831**	0.9665	0.9831	**0.9747**
Ada- Boost	0.7929	0.9693	0.7829	0.8178
RF30,1 m , 0.01*	0.87	0.68	-	0.77
Decision Tree*	0.95	0.96	-	0.88
DDoS Attack				
Algorithm	***Accuracy***	***Precision***	***Recall***	***F1 Score***
SGDC	0.9725	0.9458	0.9725	0.9590
Ada- Boost	**0.9931**	0.9929	0.9931	**0.9929**
RF30, 1 m, 0.01*	0.73	0.37	-	0.30
Decision Tree*	0.81	0.94	-	0.33
Spam Attack				
Algorithm	***Accuracy***	***Precision***	***Recall***	***F1 Score***
SGDC	0.9839	0.9680	0.9839	0.9759
Ada- Boost	**0.9873**	0.9850	0.9873	**0.9849**
RF30, 1m, 0.01*	0.94	0.99	-	0.95
Decision Tree*	0.95	0.99	-	0.96

*represents results compared from reference number [7].

CONCLUSION

In this paper, two ensemble machine learning algorithms are used to improve the capability to predict the botnet activity in 3 scenarios. In the first scenario of IRC attack is detected by the SGDC model, which performs with good accuracy of 98.31%. In the second scenario of the DDoS attack, prediction achieved is 99.31 % accuracy by the Ada-Boost model. In the third scenario SPAM attack has predicted 98.73% using Ada-boost model. Compared to conventional random forest machine learning algorithm, the attempted SGDC classification algorithm shows an 11% improvement in terms of accuracy & 10% improvement of F1 score in IRC attack. In comparison with decision tree classifier, here is 3% and 9% increases performance in terms of accuracy and F1 score respectively. In DDoS attack scenario, both algorithms improved more than 15 % accuracy and 60 % improvement in F1 score. It has achieved better prediction rate. In Spam attack scenario, both algorithms performed comparably better than the conventional machine learning algorithms. In the future, this will be applied to the deep-learning model to train with real-time attack environment for better prediction.

REFERENCES

1. N. Miloslavskaya and A. Tolstoy, "Ensuring Information Security for Internet of Things," *2017 IEEE 5th International Conference on Future Internet of Things and Cloud*, pp. 62–69, 2017.
2. G. Kambourakis, C. Kolias, and A. Stavrou, "The Mirai botnet and the IoT Zombie Armies,"

Proc. - IEEE Millinium Communication Conference MILCOM, vol. 2017-Octob, pp. 267–272, 2017.

3. I. Yaqoob *et al.*, "The rise of ransomware and emerging security challenges in the Internet of Things," Journal of *Computer Networks*, vol.129 (2017), pp. 444-458, 2017.
4. E. Bertino and N. Islam, "Botnets and Internet of Things Security," *Computer (Long. Beach. Calif).*, vol. 50, no. 2, pp. 76–79, 2017.
5. J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," *Proceeding of International Conference on IoT Social, Mobile, Analytic and Cloud, I-SMAC 2017*, pp. 32–37, 2017.
6. S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Journal of *Computer and Security*, vol. 45, pp. 100–123, 2014.
7. R. Kozik, "Distributed System for Botnet Traffic Analysis and Anomaly Detection," *Proc. - 2017 IEEE Int. Conf. Internet Things, IEEE Green Computing Communication IEEE Cyber, Physcial Social Computing IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData*, vol. 2018-January, pp. 330–335, 2018.
8. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Journal of *Computer and Security*, vol. 28, no. 1–2, pp. 18–28, 2009.
9. H. R. Zeidanloo and A. A. Manaf, "Botnet command and control mechanisms," *2009 International Conference on Computer and Electrical Engineering ICCEE 2009*, vol. 1, pp. 564–568, 2009.
10. G. Gu, J. Zhang, and W. Lee, "BotSniffer : Detecting Botnet Command and Control Channels in Network Traffic," *Proceeding of 15th Annual Network and Distributed System Security Symposium*, vol. 53, no. 1, pp. 1–13, 2008.
11. M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," *2014 International Conference on Computerand Network Communication ICNC 2014*, pp. 797–801, 2014.
12. C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using machine learning techniques to identify botnet traffic," *Proceeding on Conference of Local Computuer Networks, LCN*, no. 1, pp. 967–974, 2006.
13. J. Zhang, M. Zulkernine, and A. Haque, "Random-Forests-Based Network Intrusion," *MAN Cybern.*, vol. 38, no. 5, pp. 649–659, 2008.
14. A. S. Arunachalam, S. V. Sree, and K. Dharmarajan, "Malware Detection and Classification using Random Forest and Adaboost Algorithms," *International Journal on Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 2863–2868, 2019.
15. S. Park, I. Gondal, J. Kamruzzaman, and J. Oliver, "Generative malware outbreak detection," *Proceedings on IEEE International Confonference on Industrial Technology*, vol. 2019-February, pp. 1149–1154, 2019.
16. T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," *Proceedings, Twenty-First International Conference on Machine Learning ICML 2004*, pp. 919–926, 2004.
17. R. Kozik and M. Choraś, "Pattern Extraction Algorithm for NetFlow-Based Botnet Activities Detection," Journal of *Security and Communication Networks*, vol. 2017, pp. 1–10, 2017.
18. Dwight B. Davis, "Symantec Internet threat report - Internet of Things Cyber: ISTR 2019", April 2019. https://www.symantec.com/blogs/expert-perspectives/istr-2019-internet-things-cyber-attacks-grow-more-diverse

19. Toney Bradley,"Cyber Threat report" SonicWall on July 2019. https://www.refirmlabs.com/the-current-state-of-iot-security-sucks/

20. Mikhail Kuzin, Yaroslav Shmelev, Vladimir Kuskov,"Article on Press Releases and News by Kaspersky 2018. https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/

AUTHORS PROFILE

Santhadevi D is a Senior Research Fellow in the Department of Computer Applications, National Institute of Technology, Tiruchirappalli. She received her Master of Engineering in Computer Technology and Applications from University of Delhi and Bachelor of Engineering in Computer Science and Engineering from University of Madras. Her area of research includes in the field of cyber security with particular interest in intelligent threat detection in Internet of Things. Her contact id: santhadevi@gmail.com

Dr.B.Janet is an Assistant Professor in the Department of Computer Applications, National Institute of Technology, Tiruchirappalli for over 10 year. She has received honors that include University Rank, NET for Lectureship by UGC and deployed a honey pot sensor as part of National Cyber Coordination Centre Project for Cyber Threat Intelligence Generation. She has published more than 60 research papers in International Journals, Conferences and Book Chapters of repute in the field of Data Science. She has set up the first of its kind information Processing and Security Laboratory with Industry involvement. She is a champion of open source technology. Her areas of specialization are Information Processing and Security, Internet of Things, Application Development and Deep Learning. Her contact id: janet@nitt.edu

ISSN: 2005-4238 IJAST