Identification of Botnet Attacks using Hybrid Machine Learning Models

Amritanshu Pandey[1], Sumaiya Thaseen[2] and Ch.Aswani Kumar[3]

[1,2 & 3] School of Information Technology and Engineering, Vellore Institute of Technology,
Vellore, Tamil Nadu, India
sumaiyathaseen@gmail.com

Abstract.

Botnet attacks are the new threat in the world of cyber security. In the last few years with the rapid growth of IoT based Technology and networking systems connecting large number of devices, attackers can deploy bots on the network and perform large scale cyber-attacks which can affect anything from millions of personal computers to large scale organizations. Hence, there is a necessity to implement countermeasures to overcome botnet attacks. In this paper, three hybrid models are proposed which are developed by integrating multiple machine learning algorithms like Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), K-Nearest Neighbor (KNN) and Linear Regression (LR). According to our experimental analysis, the RF-SVM has the highest accuracy (85.34%) followed by RF-NB-K-NN (83.36%) and RF-KNN-LR (79.56%).

Keywords: Accuracy; Botnet; Classifier; Feature; Phishing.

.

1 Introduction

The process in which multiple devices are connected via internet and run an automated script for malicious intentions is called Botnet Attacks. In this kind of attack, the automated script is executed which designed to run without the knowledge of the owner of the device. Such a script is called a bot. With the advent of the Internet of Things, this is one of the most threatening concerns in the 21st Century. In the current situation, small devices like Google's Alexa and smart baby monitors are utilized to create a large- scale Botnet to perform Massive Distributed Denial of Service Attacks (DDoS). The attacks are performed by connecting thousands of IoT based devices and utilize them for targeting large scale IT systems, for example, Domain Servers and Cloud Servers. The major issue in this attack is that it is difficult to identify the source of the attack due to the integration of different devices in the network.

To build a 'botnet' or 'bot-network', 'bot-masters' need as many infected online devices or "bots" under their command as possible. The more bots are connected, the denser the botnet and the impact is huge. Bots are designed to infect millions of devices. Bot herders often deploy botnets onto computers through a Trojan horse virus. The basic strategy requires users to infect their own systems by opening email attachments, clicking on malicious pop up ads, or downloading dangerous software from a website. After being infected, botnets are free to access and modify any kind of information within the system, such as personal information. The complexity is added and they are ready to attack other computers, and commit other crimes. Complex botnets can even self-propagate; finding and infecting devices automatically. Such autonomous bots carry out seek-and-infect missions, constantly searching the web for vulnerable internet-connected devices lacking operating system updates or antivirus software. Botnets are difficult to detect. They utilize only small amounts of computing power to avoid disrupting normal device functions and alerting the user. More advanced botnets are even designed to update their behavior to thwart detection by cybersecurity software. Users are unaware that their connected device is being controlled by cyber criminals. In addition, botnet design continues to evolve, creating newer versions harder to detect by the users.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 specifies the three proposed models developed for identifying botnet attacks. Section 4 summarizes the results and section 5 concludes the work.

2 Related Work

The detection of botnet attacks has been in research for many years. In this paper[1], Adam J. Aviv and Andreas Haeberlen examined the challenges faced during evaluation of botnet detection systems. Most of the challenges arise due to difficulties in obtaining and sharing diverse sets of real network traces, as well as determining a botnet ground truth in such traces. To tackle the problem of traffic classification, Szabó et al [4] and his team devised a validation method for characterizing the accuracy and completeness of traffic classification algorithms. The important advantages of this technique is that it is based on realistic traffic mixtures, and it enables a high automation and reliable validation of traffic classification. To curb the threat of the botnet attacks, Matija Stevanovic and Jens Myrup Pedersen [20] presented a paper on contemporary botnet detection methods which utilize machine learning algorithms to identify botnet-related traffic. In this section, a comparison of existing detection methods are analyzed by comparing their characteristics, performances, and limitations. Finally, the study is concluded by showing the limitations and challenges of utilizing machine learning for identifying botnet traffic. In another study by Ping Wang and his team [8] developed a honeypot detection system. Since attackers could detect honeypots in their botnets by detecting whether the machines in a botnet can successfully send unmodified malicious traffic.. Thus a honeypot detection technique is utilized in both centralized botnets and Peer-to-Peer (P2P) structured botnets using basic detection principle. Experiments show that current standard honeypots and honeynet programs are vulnerable to the proposed honeypot detection techniques.

Different botnet identification techniques in the literature are discussed in this section. Bertino. E and N.Islam identified various botnets like Liux, Dariloz Worm and Mirai which resulted in an DDoS attack. An Intrusion Prevention System (IPS) was deployed in the network (Routers, Gateways) to monitor the traffic. R.Hallman et al

[17] detected Mirai botnet which caused DDoS attacks depending on the operational steps of the malware. M.Ozcelik et al [18] also detected Mirai botnet which propagated itself through scanning. The Mirai botnet infected the IoT devices. However, the botnets were detected by dynamic updating of flow rules where in the deployment level was called as "thin fog" and data from emulated IoT nodes and simulated network. D.H.Summerville et al [7] detected the malware by deep packet anomaly detection at the host level with two real devices. Pa et al [6] detected the DDoS attacks by implementing a honeypot to collect and analyze data at the host and network level on real dataH.Slot Edjelmaci [19] identified the routing attacks namely sink hole and selective forwarding. The authors utilized hybrid and signature based anomaly detection at the host level on simulated data. Bostani and Sheikhan [1] developed a specification based anomaly detection deployed at the network level within routers and route nodes and tested on simulation data. Midi.D et al [15] detected ICMP flood, replication worm-hole, TCP SYN flood, HELLO jamming, data modification and selective flooding. Knowledge driven anomaly detection was developed by the authors at network level on real devices and simulated data. S.Raza et al[9] identified routing attacks like spoofed or altered information sinkhole and selective forwarding.

A signature based anomaly detection technique was utilized on the simulation data at the border router and hosts. Butun et al [5] presented the challenges and opportunities in anomaly detection for IoT and cloud. The authors introduced the prominent features and application fields of IoT and Cloud. In addition, the authors also discussed security and privacy risks to personal information and finally focused on solutions from anomaly detection perspective. Zhao et al[3] proposed a new approach to identify botnets based on traffic behavior analysis using machine learning techniques. The classification is done on regular time intervals. Elisa and Nayeem [16] identified the distributed denial-of-service attacks and implemented scalable security solutions optimized for the IoT ecosystem. In spite of the anomaly-based approach's appeal [2], the industry generally favors signature-based detection for mainstream implementation of intrusion-detection systems. While a variety of anomaly-detection techniques have been proposed, adequate comparison of these methods' strengths and limitations that can lead to potential commercial application is difficult.

In the below section, we discuss some guidelines for defending against general honeypot-aware attacks. With the advent of the Internet of Things (IoT) Technology, botnet attacks have become easier and deadlier to networks and computer systems. Rohan Doshi et al., [10] demonstrated how IoT based systems can work as a botnet network to perform DDoS attacks. To protect from such attacks, his team demonstrated that using IoT-specific network behaviors to perform feature selection, DDoS can be detected in IoT network traffic with a high accuracy. The authors utilized machine learning algorithms, including neural networks to detect such attacks. Nazrul Hoque et al., [12] analyzed how DDoS attacks are built, performed, and executed. Another kind of Botnet Attack, called the HTTP botnet attack has been a popular research study. Rudy Fadhlee and Mohd Dollah [11] proposed to utilize machine learning classifiers to detect HTTP botnets. The authors achieved an accuracy of 92.93%. Nowa days, the mobile devices have wireless carriers, capable of transmitting any kind of information via multiple mediums (Bluetooth, Wi-Fi, Infrared). Shahid Anwar et al., [13] discusses about how mobile devices can be used to create a botnet and perform malicious activities. The authors conducted research on devices which had Android as their Operating System. While researchers have utilizing algorithms like Naïve Bayes and Decision trees [14], further studies are explored in the domain of deep learning where Neural Networks can be trained to detect and identify botnets. Hayretdin Bahşi et al[14] used deep learning algorithms to study how neural networks can be utilized to detect and stop botnet attacks. One of the techniques used by the authors is Dimensionality Reduction.

3. Proposed Model

3.1 RF-SVM Model

The first hybrid model, which is developed by integrating Random Forest and SVM is shown in figure 1. The RF model will be used to train the dataset by dividing into multiple units wherein each subunit is bagged to obtain the final decision unit. Every subunit of RF is reclassified by the SVM Classifier to increase the accuracy of the model. A bagging is performed to merge the results of SVM and RF. The motivation to choose these algorithms for building a hybridized model is that from our analysis, the individual classifiers namely the SVM classifier and the Random Forest model resulted in a higher accuracy F-measure, and precision compared to other supervised classifiers. The dataset is imported and stored in a table format. Before the data is split into training and testing datasets, a pre-processing operation is performed to identify which features are suitable for generating the model. The Correlation Matrix Feature Selection method is utilized to identify features having a positive correlation with the class value. The records are selected randomly from the table to split further into training and testing data. The training labels for training data i.e., the features which were selected in the previous pre-processing procedure. The training labels and the training data are used to train the RF classifier. In the RF classifier, the bagging process breaks the training data into multiple units which are basically multiple decision trees. These ‘units', are further broken down into multiple subunits, where the SVM classifier is deployed on each of the ‘sub-units'. The sub-units are then used to train the SVM classifier. The SVM Classifier has three stages; Feature Extraction, Feature Selection and Classification. Once the SVM classifies each of the subunits, a voting technique is used to perform a decision. The decision is based on majority rule. The class value which has the highest vote will be the result of the ‘Decision Fusion’ from the SVM Classifier. This is the output of one single unit of the Random Forest Classifier. The ‘Final Decision Output’ is also based on the Majority Rule. The class value which has the highest number of votes among the units will be the result of the final decision unit. Finally, from the ‘Output Unit’, the RF-SVM Hybrid model is trained and can be tested using test data. The trained RF-SVM hybrid model is utilized to predict the accuracy, precision and F1 values on the testing data.

3.2 RF-NB-KNN Model

Another hybrid model is developed is by integrating Random Forest, Naive Bayes and KNN, for detecting botnet attacks in the network. The proposed model is shown in figure 2. The RF model will be used to train the dataset by dividing into multiple units wherein each subunit is bagged to obtain the final decision unit. Every subunit of RF is reclassified by the Naive Bayes and k-NN Classifier to increase the accuracy of the model. A bagging is performed to merge the results of the base learners (Naive Bayes and k-NN) and RF. The motivation to select these algorithms for building a hybridized model is in our analysis is that the individual classifiers namely the k-NN, Naïve Bayes classifier and the Random Forest model resulted in a higher accuracy, F-measure, and precision. The entire process of preprocessing and feature selection is similar to the above model. The training labels and the training data are used to train the RF classifier. In the RF classifier, the bagging process breaks the training data into multiple units which are basically multiple decision trees. These ‘units', are further broken down into multiple subunits, where the Naive Bayes classifier is utilized on each of the ‘sub-units'.

3.3 RF-KNN-LR Model

The third novel hybrid model is developed by integrating Random Forest, KNN, and Linear Regression for detecting botnet attacks in the network. The proposed model is shown in figure 3. Similar to the previous models, RF model will be used to train the dataset by dividing into multiple units wherein each subunit is bagged to obtain the final decision unit. Every subunit of RF is reclassified by the k-NN and Linear Regression Classifier to increase the accuracy of the model. A bagging is performed to merge the results of the base learners (k-NN and linear regression) and RF. The reason to choose these algorithms for building a hybridized model is from our analysis, the individual classifiers namely the k-NN, linear regression classifier and the Random Forest model resulted in a higher accuracy, F-measure, and precision. The preprocessing and feature selection process remains same as in the earlier models. The training labels and the training data are used to train the RF classifier. In the RF classifier, the bagging process breaks the training data into multiple units which are basically multiple decision trees. These ‘units', are further broken down into multiple subunits, where we use the naïve bayes classifier on each of the ‘sub-units'. The subunits are then used to train the linear regression classifier. Within the process of the linear regression Classifier, we run the training dataset through a k-NN classifier to segregate the packet information based on whether it is a botnet or not. If it is a botnet, it will be further classified under the botnet attack category. In addition, the linear regression classifies the set of packets under the category of botnets and uses a voting technique to obtain a decision. The decision is based on whether it has been classified as a botnet or not. The class value which is the botnet category will become the output of the ‘Decision Fusion’ from the linear regression-k-NN classifier. This is the output of one single unit of the Random Forest classifier. The ‘Final Decision Output’ counts the type of botnets identified and tags the packets which look suspicious or malicious.

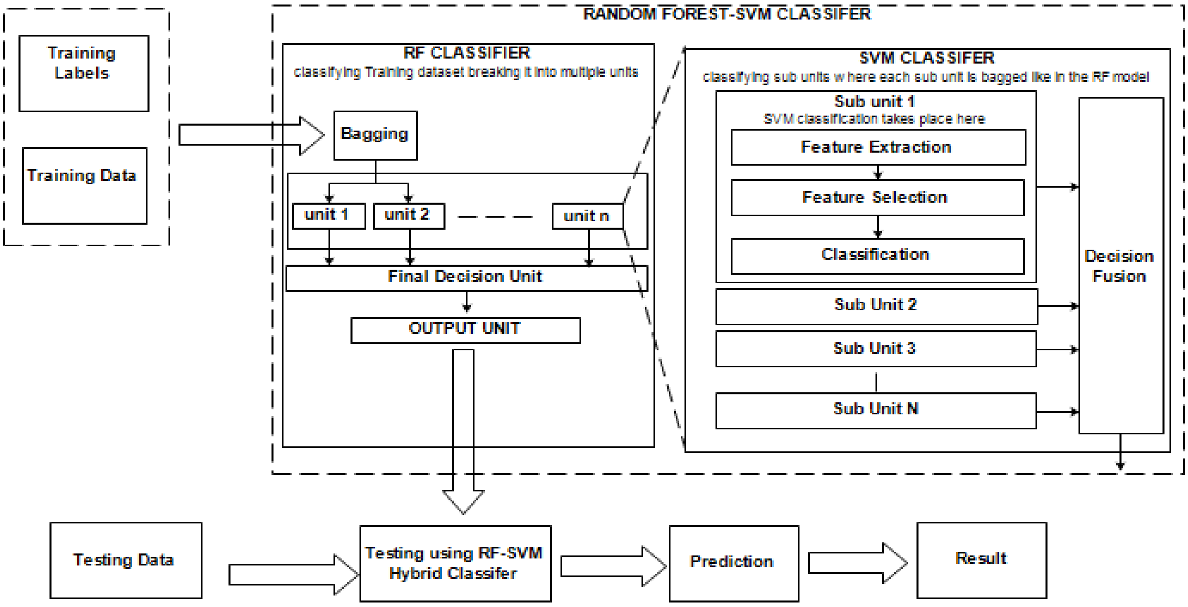

Figure 1: Proposed RF-SVM Model

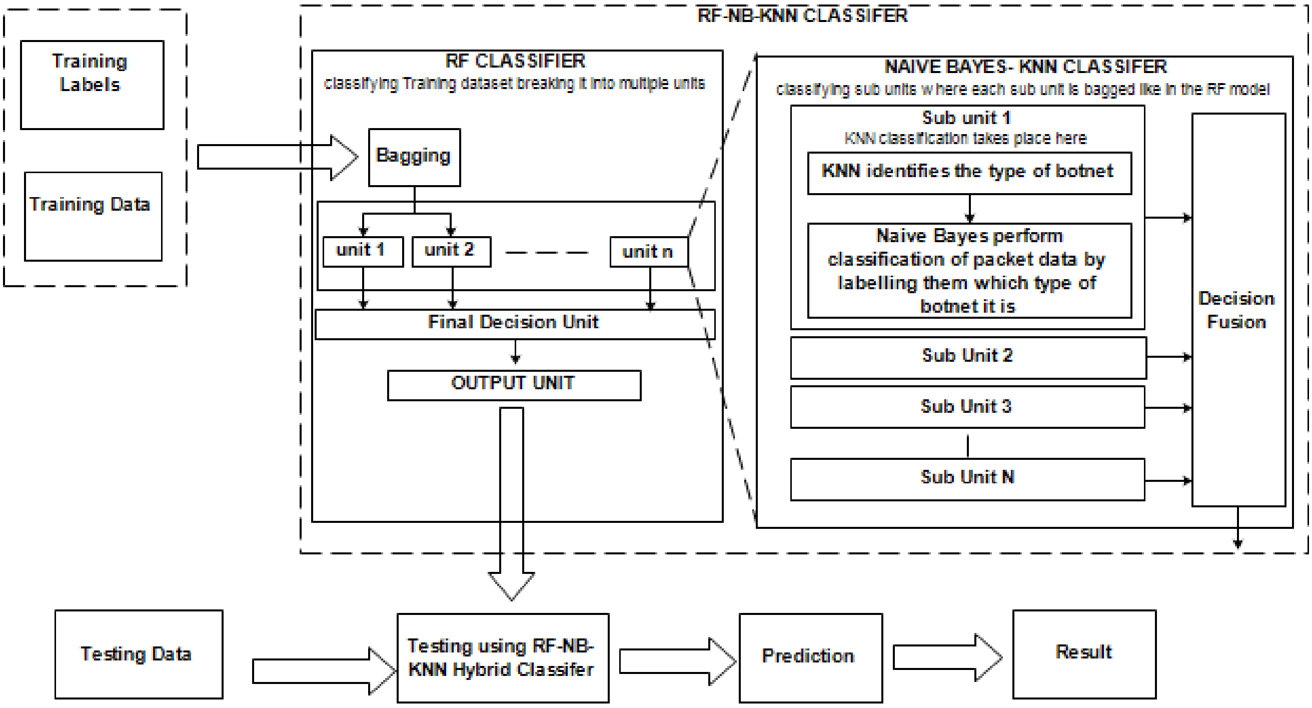
Training Labels
Training Data
RF-NB-KNN CLASSIFER
RF CLASSIFIER
classifying Training dataset breaking it into multiple units
Bagging
unit 1
unit 2
unit n
Final Decision Unit
OUTPUT UNIT
NAIVE BAYES- KNN CLASSIFER
classifying sub units where each sub unit is bagged like in the RF model
Sub unit 1
KNN classification takes place here
KNN identifies the type of botnet
Naive Bayes perform classification of packet data by labelling them which type of botnet it is
Sub Unit 2
Sub Unit 3
Sub Unit N
Decision Fusion
Testing Data
Testing using RF-NB-KNN Hybrid Classifer
Prediction
Result

Figure 2: Proposed RF-NB-KNN Model

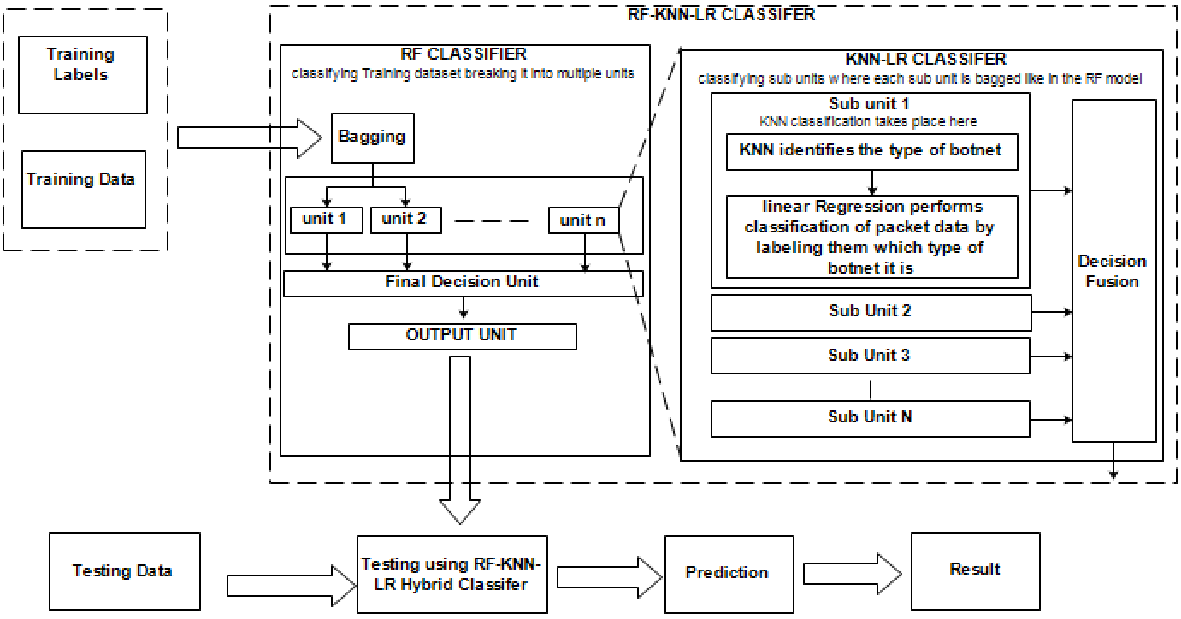
RF-KNN-LR CLASSIFER
Training Labels
Training Data
RF CLASSIFIER
classifying Training dataset breaking it into multiple units
Bagging
unit 1
unit 2
unit n
Final Decision Unit
OUTPUT UNIT
KNN-LR CLASSIFER
classifying sub units where each sub unit is bagged like in the RF model
Sub unit 1
KNN classification takes place here
KNN identifies the type of botnet
linear Regression performs classification of packet data by labeling them which type of botnet it is
Sub Unit 2
Sub Unit 3
Sub Unit N
Decision Fusion
Testing Data
Testing using RF-KNN-LR Hybrid Classifer
Prediction
Result

Figure 3: Proposed RF-KNN-LR Model

4 Experimental Results

4.1 Dataset

The dataset used to perform the comparative analysis of the hybrid algorithms is the ISCX 2012 dataset. The features used in this dataset are Time, Source, Destination, Protocol, length and Info. Here, Source and Destination show the IP Address. Time specifies the time taken by the packets to be transmitted. The type of protocol used to transmit data is specified in the protocol attribute. Info shows the information passed by the packets. The dataset consists of 5,114,514 packets out of which 80% of these packets are used for training and 20% is used for testing. This dataset has been generated in a physical testbed implementation using real devices that generate network traffic with all the standard protocols. Table 1 shows the different categories of botnets in the dataset.

4.2 Comparative Study

Table 2 shows the performance of the hybrid algorithms with regard to various metrics such as Accuracy, Precision, F1, Area under the Curve (AUC) and Classification Error. The metrics are calculated as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \; - - - - - - \; (1)$$

$$Precision(\mathrm{P}) = \frac{TP}{TP + FP} - - - - - - - - \; (2)$$

$$Recall(\mathrm{R}) = \frac{TP}{TP + FN} \; - - - - - - - - - \; (3)$$

$$F1 = \frac{2 * P * R}{P + R} ------ (4)$$

$$Classfication\ error = 100 - Acc ----(5)$$

Where, TP, --TN, FP and FN denote true Positives, True Negatives, False positives and False Negatives respectively. From table 2, it is inferred that RF-SVM has the highest Accuracy with the value of 85.34% followed by RF-Naive Bayes-K-NN with an accuracy of 83.36% and RF-k-NN-Linear Regression with an accuracy of 79.56%. In terms of Precision, the highest value of 82.78% is obtained by RF-SVM, while the lowest value is 74.35% is obtained in RF-KNN-LR. RF-Naive Bayes-KNN has the highest value for F1 with a value of 80.45% while the RF-SVM has the lowest value of 73.56%. Highest value for AUC is given by RF-SVM technique with 84.35%, while the RF-Naïve Bayes-KNN method resulted in 81.56% and RF-KNN-LR obtained a value of 20.44%. Finally, in terms of Classification Error, RF-SVM has the least classification error. Figure 7 shows the comparison of various performance metrics namely accuracy, precision, F1, AUC and classification error.

Table 1: Categories of Botnet in the Dataset

Botnet Name	Protocol	No.of Packets	Percentage
Neris	IRC	21159	12
Rbot	IRC	39316	22
Virut	HTT	1638	0.94
NSIS	P2P	4336	2.48
SMTP	Spam	11296	6.48
Spam Zeus	P2P	31	0.01
Zeus Control	C&C	20	0.01

Table 2: Performance Comparison of various Hybrid Algorithms

Performance Metrics	RF-SVM (In Percentage)	RF-Naive Bayes-KNN (In Percentage)	RF-KNN-LR (In Percentage)
Accuracy	85.3	83.36	79.56
Precision	82.7	80.45	74.35
F1	73.5	76.67	74.67
Classification Error	14.66	16.64	20.44

5 Conclusion

A serious issue in the domain of internet is there are no security measures to identify the botnets which can send malicious packets or script and enter in to the users system. In this paper, three hybrid models have been proposed and developed namely RF-SVM, RF-NB-KNN and RF-KNN-LR. The various performance metrics analyzed are classification accuracy, error, AUC, precision and recall. Among all the three models analyzed, RF-SVM proves to be superior in majority of the metrics namely accuracy, precision, AUC and Classification error. Thus RF-SVM outweighs the other models. The ensemble models prove to be better in comparison to single classifiers Hence, an integration of supervised classifiers are chosen for achieving better performance.

Acknowledgement

This research was undertaken with the support of the Scheme for Promotion of Academic and Research Collaboration (SPARC) grant SPARC/2018-2019/P616/SL "Intelligent Anomaly Detection System for Encrypted Network Traffic".

References

[1] Bostani, Hamid, and Mansour Sheikhan. "Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach." *Computer Communications, Vol.* 98,52-71,2017.
[2] Tavallaee, Mahbod, Natalia Stakhanova, and Ali Akbar Ghorbani. "Toward credible evaluation of anomaly-based intrusion-detection methods." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)Vol.* 40,No.5 ,pp.516-524,2010.
[3] Zhao, David, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, and Dan Garant. "Botnet detection based on traffic behavior analysis and flow intervals." *Computers & Security* Vol.39, pp.2-16, 2013.
[4] Szabó, Géza, Dániel Orincsay, Szabolcs Malomsoky, and István Szabó. "On the validation of traffic classification algorithms." In *International Conference on Passive and Active Network Measurement*, Springer, Berlin, Heidelberg, 2008, pp. 72-81.
[5] Butun, Ismail, Burak Kantarci, and Melike Erol-Kantarci. "Anomaly detection and privacy preservation in cloud-centric Internet of Things." In *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 2610-2615..
[6] Pa, Yin Minn Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. "Iotpot: A novel honeypot for revealing current iot threats." *Journal of Information Processin, Vol.* 24, no. 3, pp.522-533,2016.
[7] Summerville, Douglas H., Kenneth M. Zach, and Yu Chen. "Ultra-lightweight deep packet anomaly detection for Internet of Things devices." In *2015 IEEE 34th international performance computing and communications conference (IPCCC)*, 2015, pp. 1-8.

[8] Wang, Ping, Lei Wu, Ryan Cunningham, and Cliff C. Zou. "Honeypot detection in advanced botnet attacks." *International Journal of Information and Computer Security* , Vol.4, no. 1,pp. 30-51, 2010.

[9] Raza, Shahid, Linus Wallgren, and Thiemo Voigt. "SVELTE: Real-time intrusion detection in the Internet of Things." *Ad hoc networks* ,Vol.11, no. 8, pp. 2661-2674, 2013.

[10] Doshi, Rohan, Noah Apthorpe, and Nick Feamster. "Machine learning ddos detection for consumer internet of things devices." In *2018 IEEE Security and Privacy Workshops (SPW)*,2018, pp. 29-35.

[11] Dollah, Rudy Fadhlee Mohd, M. A. Faizal, Fahmi Arif, Mohd Zaki Mas'ud, and Lee Kher Xin. "Machine learning for HTTP botnet detection using classifier algorithms." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* Vol.10, no. 1-7,pp.27-30, 2018.

[12] Hoque, Nazrul, Dhruba K. Bhattacharyya, and Jugal K. Kalita. "Botnet in DDoS attacks: trends and challenges." *IEEE Communications Surveys & Tutorials, Vol.* 17, no. 4, pp.2242-2270,2015.

[13] Anwar, Shahid, Jasni Mohamad Zain, Zakira Inayat, Riaz Ul Haq, Ahmad Karim, and Aws Naser Jabir. "A static approach towards mobile botnet detection." In *2016 3rd International Conference on Electronic Design (ICED)*, 2016, pp. 563-567.

[14] Bahşi, Hayretdin, Sven Nõmm, and Fabio Benedetto La Torre. "Dimensionality reduction for machine learning based iot botnet detection." In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 1857-1862.

[15] Midi, Daniele, Antonino Rullo, Anand Mudgerikar, and Elisa Bertino. "Kalis—A system for knowledge-driven adaptable intrusion detection for the Internet of Things." In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 656-666.

[16] Bertino, Elisa, and Nayeem Islam. "Botnets and internet of things security." *Computer, Vol.* 2,pp.76-79,2017.

[17] Hallman, R., Bryan, J., Palavicini, G., Divita, J., & Romero-Mariona, J. IoDDoS-The Internet of Distributed Denial of Sevice Attacks-A Case Study of the Mirai Malware and IoT-Based Botnets. In *IoTBDS,2017,* (pp. 47-58).

[18] Özçelik, Mert, Niaz Chalabianloo, and Gürkan Gür. "Software-defined edge defense against IoT-based DDoS." In *2017 IEEE International Conference on Computer and Information Technology (CIT)*, 2017, pp. 308-313.

[19] Sedjelmaci, Hichem, Sidi Mohammed Senouci, and Mohamad Al-Bahri. "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology." In *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1-6.

[20] Stevanovic, M., & Pedersen, J. M. (2014, February). An efficient flow-based botnet detection using supervised machine learning. In *2014 international conference on computing, networking and communications (ICNC),2014,* (pp. 797-801.