Smart Prediction System for Classifying Mirai and Gafgyt Attacks on IoT Devices

Rawan Aldawod
College of Computer & Information Science
Princess Nourah bint Abdulrahman University, Riyadh,
Kingdom of Saudi Arabia
rawan.aldawod@gmail.com

Noura Alsaleh
College of Computer & Information Science
Princess Nourah bint Abdulrahman University, Riyadh
Kingdom of Saudi Arabia
Noura.e.alsaleh@gmail.com

Nojod Aldalbahi
College of Computer & Information Science
Princess Nourah bint Abdulrahman University, Riyadh
Kingdom of Saudi Arabia
Nojod.f.Aldalbahi@gmail.com

Reema Alqahtani
College of Computer & Information Science
Princess Nourah bint Abdulrahman University, Riyadh
Kingdom of Saudi Arabia
Reema1999rq@gmail.com

Sapiah Sakri
College of Computer & Information Science
Princess Nourah bint Abdulrahman University, Riyadh
Kingdom of Saudi Arabia
SBSakri@pnu.edu.sa

***Abstract*— The proliferation of Botnet attacks on IoT devices indicates that IoT network traffics is more vulnerable than other IT-based device network traffic. Mitigating this threat has led to new techniques for identifying attacks initiated by infected IoT-based devices. The study proposed an intelligent system (Maltect) that could predict botnet attacks on IoT-based devices by utilizing the machine learning model as the prediction engine. During the implementation process, there are two main parts; firstly, to perform the model classification using machine learning (ML) algorithms to determine the best-fitted model, and second, to develop the web-based system. For model classification, the study deployed the N-BaIoT dataset to train two renowned superior performance classifiers based on previous studies. Support-Vector Machine (supervised learning algorithm) and Extreme Gradient Boosting (ensemble learning algorithm) were the deployed classifiers. The study evaluated the models based on the Accuracy, Precision, Recall, and F-measure performance metrics for each attack type according to three types of IoT devices: the doorbell, security camera, and thermostat. The results denote that Extreme Gradient Boosting was the most performing model, achieving 99.9% accuracy in predicting attacks on all the IoT devices.**

Keywords*— *Botnet, extreme gradient boosting, gafgyt, Internet of Things (IoT)-based devices, mirai, support vector machines.

I. INTRODUCTION

A collection of compromised host machines is known as a botnet that carries out harmful actions. Host devices include desktop PCs, cellphones, laptops, and tablets [1][2]. Three elements make up a botnet: a "botmaster," a server to execute "command and control" known as (C&C), and a compromised computer known as a "bot" [3][4][5]. A C&C channel, such as Internet Relay Chat (IRC), Hypertext Transfer Protocol (HTTP), or peer-to-peer (P2P), is needed by the "botmaster" who directs bots and plans destructive attacks. These channels might be either centralized or decentralized, depending on the network protocols.

IoT devices grew more popular because of their ability to collect and analyze data [6][7][8]. They have been established as a significant pillar of Industry 4.0, resulting in the high implementation of IoT worldwide. The growing number of IoT devices has made them a popular target for attackers. Another problem is that cybersecurity measures for IoT devices are currently poorly understood and routinely disregarded. IoT devices, for example, commonly use weak passwords and transmit network traffic that is not secured [9][10].

Furthermore, they cannot operate complicated security solutions due to their low processing capabilities. As a result, attackers use vulnerable IoT devices to carry out various other attacks, namely distributed denial of service (DDoS) on multiple targets by inserting malicious software (malware) [11]. The DNS provider Dyn, for example, was hit by one of the most powerful DDoS attacks ever recorded, with a throughput of 1.2 terabits per second [12]. The Mirai, a malware that carries out attacks, spreads across IoT devices to form a botnet.

Malware-designed botnets infiltrate as many machines as possible, publicize themselves and modify their habits to find and attack devices automatically. As a result, it is quite challenging to recognize botnets. The fact that botnets hide on devices with little effect on their functionality makes them extremely hard to identify and contain. A security camera, for example, could be a part of an active botnet that is not aware of neither by the average user or a small business. As a result, detecting botnets in IoT device traffic is critical. Because IoT devices are plentiful in various settings, malicious actors exploit them as a powerful and mysterious playground. Their main goal is to build botnets to fulfill their criminal purposes, which include spam, advertisement fraud, and DDoS attacks. The attack is intricate and varied. The present detection tools may take too long to detect and are not very accurate. All of the IoT devices were infiltrated by Botnets before they could finish the detection analysis, and it was too late to stop the attack.

979-8-3503-2028-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00274

This paper is organized into six sections; section 1 introduces the research; section 2 elaborates on previous works related to this study; section 3 presents the methodology; section 4 discusses the experimental results of the study; and section 5 concludes the study, including some suggestions for future work.

II. Related Work

A. Botnets Attack on IoT Devices

Without this new vulnerable IoT equipment, securing the Internet is difficult enough [13]. They can break into any internet-connected gadget, including smartwatches and corporate machines. Botnets like Gafgyt and Mirai have open-source code [13]. Mirai is a self-propagating botnet that may be used to turn a machine into a "zombie" that can then be controlled from a distance. These actions resulted in a significant rate of cyberattacks, primarily on Internet of Things (IoT) devices like security cameras, doorbells, web cameras, etc. Mirai regularly checks the infected IoT device's IP address using the default log-in credentials. Five known malware executed by Mirai is scan, ack, syn, UDP, and udpplain. Gafgyt is a botnet that targets Linux machines and launches DDoS assaults. This botnet uses five attack methods: combo, junk, scan, UDP, and TCP. Combo sent spam-filled material and revealed the IP address and port connection. This attack is known as spam. Scan searches for weak points in a network. UDP and TCP cause request flood to happen. This botnet's descendants and imitations educated the industry to be cautious while handling IoT devices [14]. Botnets have infected a large number of devices. More complex botnets can even detect and infect devices on their own. Botnets are hence uncommon [15]. Botnets are also challenging to detect and block since they hide on machines that aren't performing well [15]. For example, a small business or individual may not realize that a security camera is part of a botnet. As a result, scanning IoT traffic for botnets is crucial.

B. Botnet Detection Methods Classification

Approaches to bot detection and prevention have piqued people's interest. The most critical criterion in botnet detection is identifying infected workstations before botnets abuse them to initiate harmful actions. Researchers discussed several methods for detecting botnets in previous studies. References [16, 17] explored three botnet detection methods: intrusion detection systems (IDS), IDS-based, and Honeynet. IDS detection includes systems that are signature-based and anomaly-based. For anomaly-based detection, there are three approaches: hybrid-based, host-based, and network-based. Other studies [10][18][19][20] have classified botnet detection strategies into four groups: IDS, DNS-based, Honeynet, and ML-based techniques. The authors [8][21][22][23][24] categorize botnet detection strategies according to the deployed ML learning model, whether supervised or unsupervised botnet detection strategies. Fig. 1 presents the classification of the Botnet detection approaches.

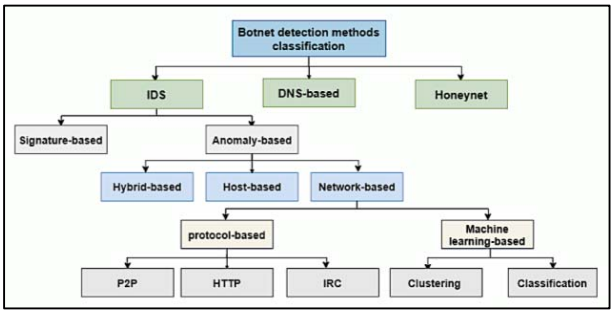

Fig. 1. Botnets Detection Technique Classification

C. ML Classification Methods

Artificial Intelligence (AI) is a field that improvises existing systems by past experience learning, and future forecasting through data analysis. ML is a sub-field of AI where ML algorithms learn from earlier experiences data, extracting patterns and building a predictive model. To test the efficiency and effectiveness of the model, anonymous data will be fed to it, which will assist in making well-informed and timely decisions concerning the current issue. There are five ML classification methods, namely supervised learning (SL), unsupervised learning (UL), reinforce learning (RL), deep learning (DL), and semi-supervised learning (SSL) [21][22][25]. SL is used in labelled datasets. The classifiers "learn" from the dataset's identified patterns and use them to anticipate future data labels. Examples of often used classifiers are Random Forest (RF), k-Nearest Neighbor (kNN), Naive Bayes (NB), and Support Vector Machine (SVM) [21][23]. UL learn from unlabeled dataset to develop the model by studying data attributes. The most often used classifiers are K-means, principal component analysis (PCA), Apriori, and Latent Dirichlet Allocation (LDA) [21][23]. RL is a learning method using feedback that operates in an environment. The classifiers that are often used are Q-learning and SARSA (State-Action-Reward-State-Action). DL is a learning-by-example technique that teaches machines to behave like humans. The frequently used classifiers are the convolutional neural network (CNN), recurrent neural networks (RNNs), generative adversarial networks (GANs), long short-term memory networks (LSTMs), and multilayer perceptron (MLPs). SSL teaches algorithms to learn from labeled data (small numbers) and unlabeled data (large numbers) [21][26]. However, SSL fails to use the mixed data effectively and needs specialized learning classifiers such as Logistic Regression / the Back Propagation Neural Network / Apriori algorithm / K-Means [22].

D. Latest Studies of Botnets Detection Using ML

In this study, reviewing the latest studies on Botnets detection utilizing ML would highlight the gap between the previous studies and the proposed study. Furthermore, reviewing the latest studies would also provide essential insights and trends of the studies in this domain. Table I presents the summarization of the studies.

TABLE I. ML-BASED BOTNETS STUDIES SUMMARY

Study/Year	Algorithms	Database	Model Performance
[27]/2011	NB+SVM+ k-NN,	Hungary Repository	Accuracy: 97%
[25]/2012	SVM+RF+DT+ J48,	University Network Traffic	True Positive Rate: 65%, False Positive Rate: 1%
[28]/2014	DT+C4.5	ISOT, ISCX,	Accuracy: 75%
[29]/2016	C4.5+DT+RT	ISOT	Accuracy: 99.9%
[30]/2017	RF	CTU	Accuracy: 93.6%
[31]/2018	J48+SVM	Botnet Samples	Accuracy: HTTP - 80%, IRC - 95%.
[32]/2018	GNB+NN+DT	CTU	F1 score = 0.99
[33]/2019	DT+KNN+LR+ ANN	CTU, ISOT	Accuracy: 98.7%
[34]/2019	CNN+LSTM	DG-Archive, OSINT	Accuracy: 97.80%
[35]/2019	DT+NB+ANN	CTU	Accuracy: 94.4%
[36]/2020	ANN+J48+DT +NB	N-BaIoT	Accuracy: 99%
[37]/2020	ANN+J48+DT +NB	N-BaIoT	Accuracy: 99%
[38]/2020	GNN	CAIDA	Accuracy: 99.5%
[39]/2020	J48+DT	Peer-Rush-botnet, ISOT, ISCX	Accuracy: 99.9%
[40]/2020	LSTM+RNN+ MCFP	Botnet Samples	Accuracy: 99.5%
[41]/2020	NB+DT	ISCX, CTU	Accuracy: 99.6%
[42]/2021	I-SVM, IForest, Adaboost	N-BaIoT	Accuracy: 99.9%

III. METHODOLOGY

Maltect's system implementation adopts the generic system development architecture, as shown in Fig. 3.

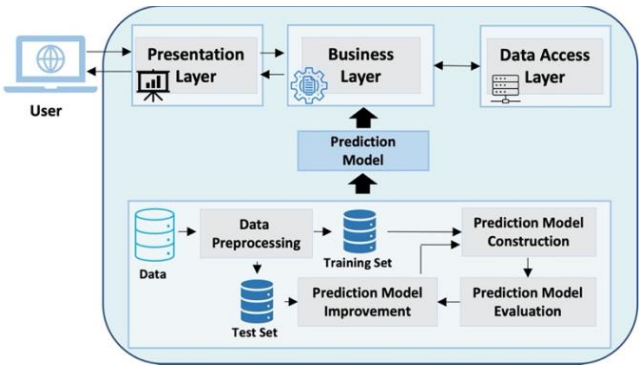

Fig. 3 Proposed Maltect's System Architecture

A. Presentation Layer

This project used python programming language, Flask library, HTML, CSS, and JavaScript to develop Maltect. The project implemented ten primary interfaces mainly interacting with the business layer for attack prediction. The following sub-section describes the interfaces.

B. Business Layer

This layer is the system's heart, which embeds the system's prediction engine. The engine can predict which Botnet has attacked which IoT device. The model classification process used Python programming. The following paragraph explains the model classification phases.

- Data Acquisition: The study acquired the N-BaIoT dataset from the University of California, Irvine (UCI) [4]. The file consists of two botnet malware datasets. Each dataset comprises five types of malware attacks on five categories of IoT devices: the doorbell, the security camera, the webcam, the baby monitor, and the thermostat. For the gafgyt botnet, the attack network traffic is identified as a combo, junk, scan, UDP, and TCP malware. While for the mirai botnet, the attack network traffic is known as ACK, Scan, Syn, UDP, and UDPPlain malware.

- Data Preprocessing: The study must first preprocess the raw data before training the classifiers. The preprocessing involves the process of data cleaning, normalization, and randomization. This process divided 70% data into a training set and 30% into a testing set.

- Prediction Model Construction: Two selected renowned classifiers, namely SVM and extreme gradient boosting (XGBoost), were trained using the training dataset. The classifiers were chosen based on their superior performance in previous studies. Table II presents the description of the classifiers.

TABLE II. DESCRIPTION OF THE DEPLOYED ML CLASSIFIERS

Classifier	Description
SVM	SVM is a supervised learning algorithm often used to solve classification and regression problems. It works by finding the best hyperplane to separate a set of classes in a dataset. To reduce errors, SVM will maximize the margin distance between separated classes as far as possible.
XGBoost	XGBoost is an ensemble learning algorithm frequently used to solve classification and regression problems during modeling. XGBoost works by tuning the prediction model mistakes by adding more decision trees to the ensemble. Hence will increase the model's accuracy performance.

- Prediction Model Evaluation: Four performance metrics, namely the classification accuracy, precision, recall, and F1-score, evaluated the models. Table III presents the description of the metrics.

TABLE III. DESCRIPTION OF THE DEPLOYED PERFORMANCE METRICS

Metrics	Description
Accuracy	Accuracy is a metric to measure how often the classifier can make correct predictions. It is a ratio between accurate predictions number against the total prediction number: • TP (True-Positives) – "True" in the actual number of attacks, and "True" in the predicted number of attacks • TN (True-Negatives) – "False" in the actual number of attacks, and "False" in the predicted number of attacks • FP (False-Positives) – "False" in the actual number of attacks, but "True" in the predicted number of attacks • FN (False-Negatives) – "True" in the actual number of attacks, but "False" in the predicted number of attacks Formula: (TP + TN) / (TP + FP + FN + TN)
Precision	The classifier specificity, also known as precision, denotes the ratio of true positives against the total positive instances. A lower sensitivity rate equals higher false negatives instances, whereas a low specificity indicates many false positives instances. Formula: TN / (FP + TN)
Recall	The classifier sensitivity, also known as recall, is the ratio of actual positive instances against the total positive samples. Formula: TP / (TP + FN)
F1-Measure	A low sensitivity ratio indicates a high number of false negatives instances, while a low specificity ratio indicates many false positives samples.

- Prediction Model Improvement: Using the test data would improve the models to ensure the prediction models' fitness.

C. Data Access Layer

The data access layer, often known as the database tier, is where the application's processed data is kept and maintained. This layer locates the database servers, which can be accessed using Oracle or MySQL and Microsoft SQL Server. It is located separately from application servers and business logic.

IV. RESULTS AND DISCUSSION

The study divided the results into two parts. Firstly, we presented the results of the model classification to determine the best-fitted model for the Maltect system, and secondly, the screenshots of the system's interface implementation.

A. Results of the Model Classification

Fig. 4 and Fig. 5 present the results of the model classification for the Botnets attack in Danmini Doorbell.

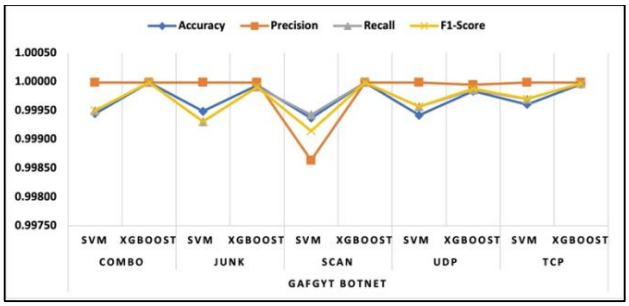

Fig. 4. Model classification results of Gafgyt attacks on Danmini Doorbell

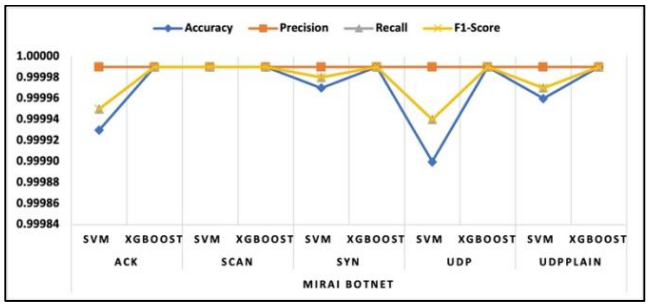

Fig. 5. Model classification results of Mirai attacks on Danmini Doorbell

Fig. 6 and Fig. 7 present the results of the model classification for the Botnets attack on Provision PT 838 Security Camera.

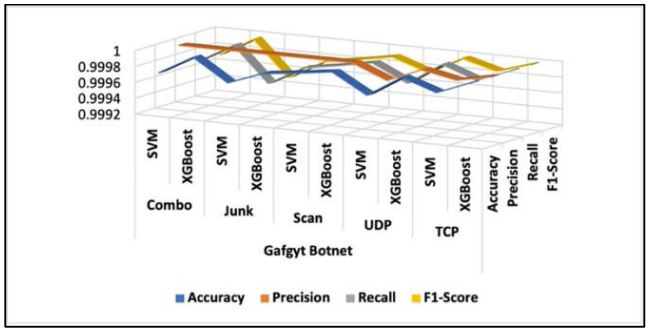

Fig. 6. Model classification results of Gafgyt attacks on Security Camera

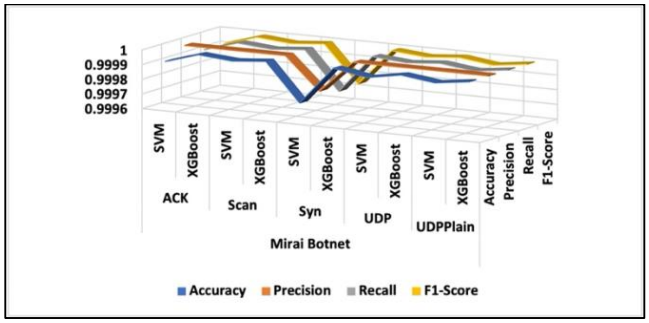

Fig. 7. Model classification results of Mirai attacks on Security Camera

Fig. 8 and Fig. 9 present the results of the model classification for the Botnets attack on the Ecobee Thermostat.

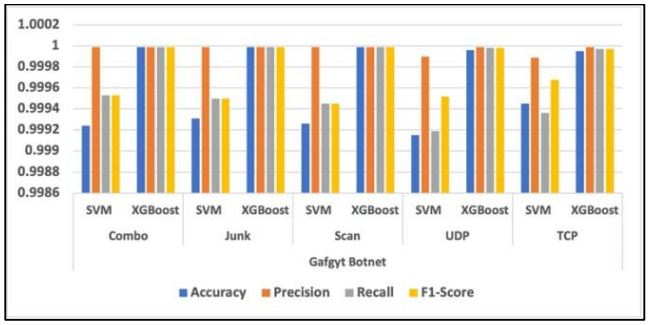

Fig. 8. Model classification results of Gafgyt attacks on Ecobee Thermostat

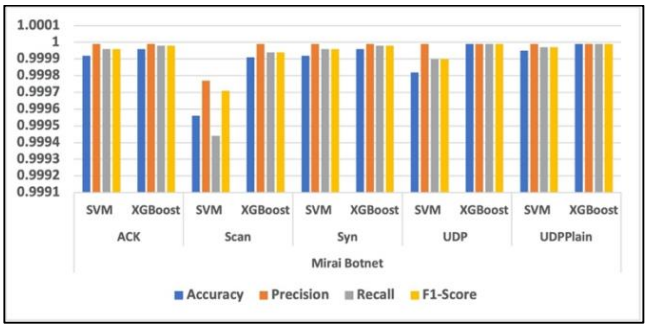

Fig. 9. Model classification results of Mirai attacks on Ecobee Thermostat

B. *Model Classification Results in Discussion*

a) Attack on Danmini Doorbell: Fig. 4 observed that for predicting the Gafgyt attack on Danmini Doorbell, the XGBoost model's accuracy achieved the highest result of 99.99% to predict the attack by TCP. The SVM model's accuracy is the lowest (99.95%) in predicting the Botnet via Scan attack. Fig. 5 denotes that for predicting the Mirai attack, the XGBoost model performs well to consistently predict the attack by ACK and Scan with a value between 99.998% - 99.999%. However, the UDP attack has the lowest accuracy value (99.990%).

b) Attack on Provision PT 838 Security Camera: Fig. 6 clearly show that the XGBoost model performs well in predicting a Gafgyt attack on Provision PT 737E – security camera device with an accuracy value of 99.98%. For predicting the Mirai attack (see Fig. 7), again XGBoost model outperformed the rest of the models with 99.99% accuracy.

c) Attack on Ecobee Thermostat: Fig. 8 and Fig. 9 indicate XGBoost consistently achieved higher accuracy of 99.99% than SVM in predicting both Gafgyt and Mirai botnet attacks on the Encobee Thermostat. However, the SVM model obtains a slightly lower accuracy value of 99.98%.

C. *Maltect System Interfaces Screenshot*

Fig.10. Screenshot of Maltect main page

Fig. 10 shows the snapshot of the system's main page. Users have two options whether to choose the log-in or sign-up page.

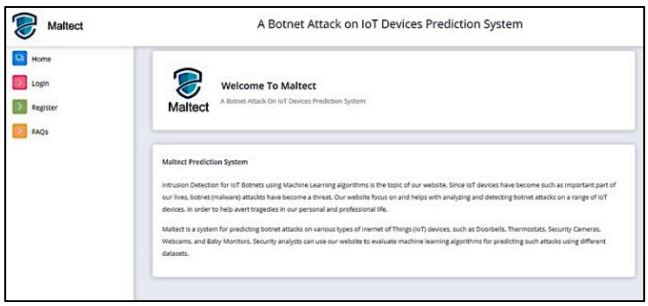

Fig.11. Screenshot of the Maltect's Home Page

Fig. 11 displays the system's home page. The system will load the home page after logging in to the user's accounts successfully. This page guides novice users to correctly used the system.

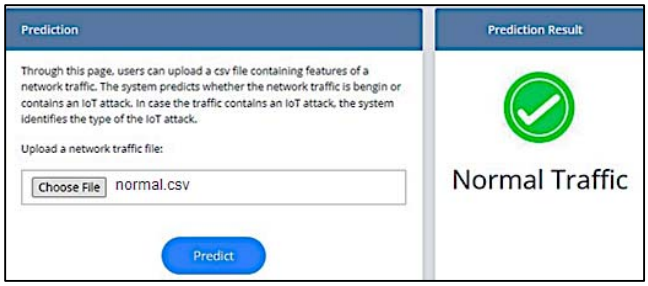

Fig. 12. Maltect's prediction interface displaying the prediction result that shows the normal traffic or no attack case

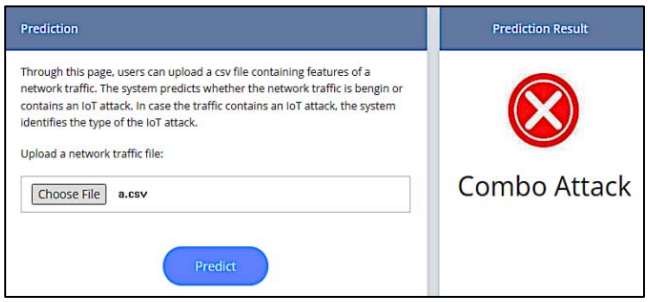

Fig. 13. Maltect's prediction interface displaying the prediction result of the combo attack identified in the uploaded network traffic file

Fig. 12 above presents the system prediction page. The system will load this page after the user clicks the predict button. The outcome of the system shows that the prediction model has successfully classified the botnet attack by learning from the IoT network traffic file uploaded for prediction.

V. Conclusion and Future Work

The project indicated the proposed system as complete after achieving all project objectives. The study reviewed similar systems to discover the current gaps. Throughout the review, one common problem found is that there is a lack of a complete system incorporating a predictive model as the outcome of the model classification techniques. In other words, the studies solely focus on these techniques without implementing an application system as a whole. Thus, the proposed system overcomes this issue by developing an exemplary user interface

for front-end and back-end systems allowing users to benefit from it.

As mentioned earlier, the project deploys supervised learning and ensemble learning models to determine the best-performing model. After training and testing the two renowned ML algorithms, the study observed that the XGBoost algorithm performs the best by obtaining 99.99% accuracy. In contrast, the SVM algorithm performs lower than XGBoost. Thus, XGBoost is being selected and used for the botnet prediction process on IoT devices.

Overall, by implementing the system, the project can gradually solve the existing problems. Organizations can save their expenditure on security expert consultation fees because they can use the system to predict the botnet attack type and undertake appropriate security countermeasures. Also, implementing machine learning within the system can help the users know the predicted types of botnet attacks within a short period. Thus, the organization can take advantage of the system as they do not need to spend time and effort getting security analyst services to access the security posture of the organization's IT infrastructure. At the same time, the system can help to increase security awareness among the users or employees of the organization.

As a whole, the system increases the chance of accessing the security level as the system can give users a lot of conveniences that can address the issues brought up in the introduction section. Moreover, novice users who do not have any security analysis experience can efficiently use this system.

The project expects to contribute significantly to the IT security domain as a complete functional prediction system compared to previous studies focusing solely on the model classification process. Thus, it provides the option for testing the feasibility of the system feasibility

For future enhancement, the system can add or improve a few functionalities. The system can be enhanced from the tester's feedback, improving the proposed system by experimenting with more new attack network traffic datasets and discovering the latest attack on various IoT device categories. This effort will allow the researcher to investigate more latest botnet types.

ACKNOWLEDGMENT

We would like to thank Eng. Muhammad Alagil and Nora Alagil for their support and generous grant in sponsoring our participation in this conference (Grant No .....). Our gratitude also goes to Princess Nourah bint Abdulrahman University for facilitating the process of participation.

REFERENCES

[1] K. Angrishi, "Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT botnets," ArXiv:1702.03681, pp. 1-17, 2017.

[2] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," Computer, vol. 50, no. 7, pp. 80-84, 2017.

[3] C. She, W. Wen, Z. Lin, and K. Zheng, "Application-layer ddos detection based on a one-class support vector machine," International Journal of Network Security Its Applications, vol. 9, pp. 13–24, 01 2017.

[4] R. Vishwakarma and A. K. Jain, "A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks,"

[5] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, "Deepdetect: Detection of distributed denial of service attacks using deep learning," Comput. J., vol. 63, pp. 983–994, 2020.

[6] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in 2018 IEEE Security

[7] UCI Dataset. UCI Machine Learning Repository: Detection_of_iot_botnet_attacks_n_baiot data set. (n.d.). Retrieved February 20, 2022, from https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[8] B. Alothman, "Similarity based instance transfer learning for botnet detection," Int. J. Intell. Comput. Res. 9, pp. 880–889, 2018.

[9] I. Ali, A.I.A. Ahmed, A. Almogren, M.A. Raza, S.A. Shah, A. Khan, A. Gani, "Systematic Literature Review on IoT-Based Botnet Attack," IEEE Access, vol. 8, pp. 212220–212232, 2020.

[10] S.S. Silva, R.M. Silva, R.C. Pinto, R.M. Salles, "Botnets: A survey," Comput. Netw. , vol. 57, pp. 378–403, 2013.

[11] R. Limarunothai, M.A. Munlin, "Trends and challenges of botnet architectures and detection techniques," J. Inf. Sci. Technol., vol. 5, pp. 51–57, 2015.

[12] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," Cybersecurity, vol. 2, pp.1–22, 2019.

[13] M. Singh, M. Singh, S. Kaur, "Issues and challenges in DNS based botnet detection: A survey," Comput. Secur., vol. 86, pp. 28–52, 2019.

[14] S. Gaonkar, N.F. Dessai, J. Costa, A. Borkar, S. Aswale, P. Shetgaonkar, "A survey on botnet detection techniques," In Proceedings of the2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 24–25 February 2020; pp. 1–6.

[15] P. Amini, M.A. Araghizadeh, R.A. Azmi, "Survey on Botnet: Classification, detection and defense. In Proceedings of the 2015 International Electronics Symposium (IES), Surabaya, Indonesia, 29–30 September 2015; pp. 233–238.

[16] H.R. Zeidanloo, M.J.Z. Shooshtari, P.V. Amoli, M. Safari, M. Zamani, "A taxonomy of botnet detection techniques," In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; vol. 2, pp. 158–162.

[17] A. Karim, R.B. Salleh, M. Shiraz, S.A.A Shah, I. Awan, N.B. Anuar, "Botnet detection techniques: Review, future trends, and issues," J. Zhejiang Univ. Sci. C 2014, vol.15, pp. 943–983,

[18] I. Ghafir, J. Svoboda, V. Prenosil, "A survey on botnet command and control traffic detection," Int. J. Adv. Comput. Netw. Secur., vol. 5, pp. 7580, 2015.

[19] J. Vania, A. Meniya, H. Jethva, "A review on botnet and detection technique" Int. J. Comput. Trends Technol., vol. 4, pp. 23–29, 2013.

[20] S. Asha, T. Harsha, B. Soniya, "Analysis on botnet detection techniques," In Proceedings of the 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS), Karnataka, India, 6–7 May 2016; pp. 1–4.

[21] S. Miller, C. Busby-Earle, "The role of machine learning in botnet detection," In Proceedings of the 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), Barcelona, Spain, 5–7 December 2016; pp. 359–364.

[22] F. Samson, V. Vaidehi, "Hybrid botnet detection using ensemble approach," J. Theor. Appl. Inf. Technol, 2017, vol.95, pp.1646–1654, 2017.

[23] T.S. Hyslip, J.M. Pittman, "A survey of botnet detection techniques by command and control infrastructure," J. Digit. Forensics Secur. Law, 2015, vol. 10, pp. 2, 2015.

[24] P. Krishnan, S. Duttagupta, K. Achuthan, "VARMAN: Multi-plane security framework for software defined networks," Comput. Commun., vol. 148, pp. 215–239, 2019.

[25] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, C. Kruegel, "Disclosure: Detecting botnet command and control servers through large-scale

netflow analysis," In Proceedings of the 28th Annual Computer Security Applications Conference, Orlando, FL, USA, 3–7 December 2012; pp. 129–138.

[26] X. Wang, Y. Xu, C. Chen, X. Yang, J. Chen, L. Ruan, Y. Xu, R. Chen, "Machine Learning Empowered Spectrum Sharing in Intelligent Unmanned Swarm Communication Systems: Challenges, Requirements and Solutions," IEEE Access, vol. 8, pp. 89839–89849, 2020.

[27] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," In Proceedings of the 2011 Ninth Annual International Conference on privacy, Security and Trust, Montreal, QC, Canada, 19–21 July 2011; pp. 174–180.

[28] E.B. Beigi, H.H. Jazi, N. Stakhanova, A.A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," In Proceedings of the 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, USA, 29–31 October 2014; pp. 247–255.

[29] O.Y. Al-Jarrah, O. Alhussein, P.D. Yoo, S. Muhaidat, K. Taha, K. Kim, "Data randomization and cluster-based partitioning for botnet intrusion detection," IEEE Trans. Cybern., vol. 46, pp. 1796–1806, 2015.

[30] R. Chen, W. Niu, X. Zhang, Z. Zhuo, F. Lv, "An effective conversation-based botnet detection method," Math. Probl. Eng., pp. 2017, 2017.

[31] M.S. Gadelrab, M. ElSheikh, M.A. Ghoneim, M. Rashwan, "BotCap: Machine learning approach for botnet detection based on statistical features," Int. J. Commun. Netw. Inf. Secur., vol. 10, pp. 563, 2018.

[32] S. Ryu, B. Yang, "A comparative study of machine learning algorithms and their ensembles for botnet detection," J. Comput. Commun., vol. 6, pp. 119, 2018.

[33] R.U. Khan, X. Zhang, R. Kumar, A. Sharif, N.A. Golilarz, M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," Appl. Sci., 2019, vol. 9, pp. 2375, 2019.

[34] R. Vinayakumar, K. Soman, P. Poornachandran, M. Alazab, A. Jolfaei, "DBD: Deep learning DGA-based botnet detection," In Deep Learning Applications for Cyber Security; Springer: Berlin/Heidelberg, Germany, 2019; pp. 127–149.

[35] R.U. Khan, R. Kumar, M. Alazab, X. Zhang, "A Hybrid Technique to Detect Botnets, Based on P2P Traffic Similarity," Technical Report; In Proceedings of the 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, VIC, Australia, 8–9 May 2019.

[36] T.A. Tuan, H.V. Long, L.H. Son, R. Kumar, I. Priyadarshini, N.T.K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," Evol. Intell., 2019, vol. 13, pp. 283–294, 2019.

[37] Y.N. Soe, Y. Feng, P.I Santosa, R. Hartanto, K. Sakurai, "Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture," Sensors 2020, vol. 20, pp. 4372, 2020.

[38] J. Zhou, Z. Xu, A.M. Rush, M. Yu, "Automating Botnet Detection with Graph Neural Networks," arXiv 2020, arXiv:2003.06344.

[39] P. Gahelot, N. Dayal, "Flow based botnet traffic detection using machine learning," In Proceedings of ICETIT 2019; Springer: Cham, Switzerland, 2020; pp. 418–426.

[40] W.C. Shi, H.M. Sun, "DeepBot: A time-based botnet detection with deep learning," Soft Comput., 2020, vol. 24, pp. 16605–16616, 2020.

[41] S. Almutairi, S. Mahfoudh, S. Almutairi, J.S. Alowibdi, "Hybrid Botnet Detection Based on Host and Network Analysis," J. Comput. Netw. Commun., pp. 2020, 2020.

[42] B. Sudharsan, D. Sundaram, P. Patel, J.G. Breslin, & M. I. Ali, "Edge2guard: Botnet attacks detecting offline models for resource-constrained IoT devices," In 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) (pp. 680-685). IEEE.