Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/18/20230966

Detection of IOT botnet attack based on machine learning

Chi Cheng[1,†], Kaiyue Zhang[2,†], Yuhang Zhang[3,4,†]

[1]Viterbi School of Engineering, University of Southern California, Shenyang, 110102, China
[2]School of Finance, Shanghai University of International Business and Economics, Shanghai, 054800, China
[3]School of Computer Science & Technology, Beijing Institute of Technology, Beijing, 102300, China

[4]yuhangzhang@bit.edu.cn
[†]These authors contributed equally.

Abstract. The paper discussed the process of data processing and algorithm selection for three different scenarios in order to improve accuracy in detecting DDOS attacks, SPAM emails, and malware. It provided detailed descriptions of each process involved in the simulation. For the DDOS attack detection simulation, three different datasets were used, and missing data was removed to ensure the quality of the data. In addition, features were processed to make sure they could be applied to specific algorithms. Both decision tree and random forest algorithms were selected and tuned to obtain maximum accuracy. Similarly, for the SPAM email detection simulation, binary was used to represent whether an email was spam or not, and Count Vectorizer function was applied to convert mail contents into feature vectors. The KNN and decision tree algorithms were chosen, and emphasis was given on parameter adjustment to eliminate overfitting and ensure optimal model accuracy. The paper also discussed the importance of considering multiple factors when selecting and tuning algorithms, such as accuracy, complexity, and computational efficiency. These factors must be balanced to achieve the best overall performance. Overall, the paper provided a comprehensive overview of the methods and processes involved in data processing and algorithm selection to improve detection accuracy for DDOS attacks, SPAM emails, and malware. This research can greatly benefit organizations that are looking to enhance their security measures and minimize the risks associated with these cyber threats.

Keywords: IOT, botnet attack, malware, machine learning, KNN, decision tree.

1. Introduction

The Internet of Things (IoT) is a network system that integrates data and information, enabling smart devices to interact with each other through wired or wireless connections. This system typically consists of six layers: coding, perception, network, middleware, application, and business. IoT systems utilize various technologies such as computer vision, radio frequency identification, wireless sensor networks, and cloud computing to provide diverse functions that can be customized for different applications. The highly modular nature of IoT makes it easy to integrate or disassemble systems as needed [1].

DOI: 10.54254/2755-2721/18/20230966

Today, the IoT has revolutionized the way humans interact with the world. With billions of interconnected devices, IoT networks have facilitated seamless communication and automation in various sectors such as healthcare, manufacturing, smart cities, and home automation. However, the rapid growth and widespread adoption of IoT have also given rise to new security challenges. Among these challenges, IoT botnet attacks have emerged as a significant threat to the integrity, confidentiality, and availability of IoT networks [2]. Botnets are a common means of attacking IoT systems, as their self-propagating approach allows them to affect the entire system on a large scale. These attacks can use a multi-layer command architecture to achieve anonymity, and each attack node may be located in a different part of the world, making monitoring and tracking botnet activity extremely difficult. In addition, botnet attacks often have a certain degree of stealth, making it challenging for users to detect if their devices have been infected or hijacked.

Detection of IoT botnet attacks has become a critical research area, as traditional security mechanisms are often ill-suited to protect the heterogeneous and resource-constrained IoT environment. Machine learning techniques have demonstrated promising potential for detecting and mitigating IoT botnet attacks, leveraging their ability to identify complex patterns and behaviors within large datasets. This paper aims to provide a comprehensive overview of state-of-the-art methods for detecting IoT botnet attacks based on machine learning [3,4]. The unique characteristics and challenges posed by IoT networks are examined, and key machine learning algorithms employed for botnet detection are reviewed. The paper also discusses evaluation metrics and datasets used to validate these approaches [5].

2. Detection method

Due to the variety and complexity of botnet attacks, there is no one way to deal with all attacks [6]. However, the communication mode between the Internet of Things system and the Internet and the cloud is relatively limited, which makes it possible to deal with botnet attacks against the Internet of things in limited ways. As mentioned before, botnet attacks can be divided into distributed denial of service attacks, spam attacks and malware attacks according to different modes of transmission. Generally, these attacks have different characteristics according to the different ways, and there may be some correlation between these features, which will be the key to identify botnets.

2.1. DDOS attack

DDOS attacks cause damage by overloading victims and preventing their network resources from running properly. In a DDOS attack, hackers often aggregate a large number of hosts to achieve more bandwidth than the victim can handle. These hosts often come from different regions, which makes filtering for specific hosts extremely difficult. However, some patterns can be found in DDOS attacks. For instance, although hackers can use different types of traffic to implement attacks, most zombies tend to use the same type of traffic during the operation. Meanwhile, the characteristics such as data type and attacked port can reflect the personal preference of the attacker to some extent. These features, which can be recognized by computers to a certain extent, can effectively improve the efficiency of machine learning training and improve the accuracy of detection [7].

2.2. SPAM

Spam is relatively difficult to identify because it cannot be identified by external characteristics such as the sender. Meanwhile, it is difficult to guarantee the complete accuracy of detection results when identifying mail content due to the great differences in grammar rules of different languages and the use of some network languages. Even tremendous tech companies like Google cannot completely avoid misclassifying regular mail as spam when processing it [8]. Fortunately for Internet of Things systems, there are only a few occasions when spam detection is necessary. Therefore, only a small representative sample of emails is analyzed to demonstrate the feasibility of machine learning to detect spam that the Internet of Things might receive.

DOI: 10.54254/2755-2721/18/20230966

2.3. Malware

The detection of malware was the most complex of the three examples tested. Compared with DDOS attacks, malware detection may not be able to collect enough features for analysis [9]. Compared with spam, the data transmitted by malware is too rich and abstract. The existence of errors seems inevitable when it comes to finding regularities and analyzing them from a finite set of features. Fortunately, closed-source systems and firewall software are sufficient to prevent the installation of malware in the vast majority of cases. Similar to the processing of spam samples, only a small number of representative samples are selected in this paper to verify the feasibility of machine learning-based malware detection [10].

3. Data processing

In order to ensure the reliability and authenticity of the data as far as possible, it is necessary to guarantee the diversity of data resources. Therefore, three datasets are selected in this process. To be specific, one dataset is from Czech Technical university, and the other two are from Kaggle.

3.1. DDOS

In the DDOS attack detection simulation, the data contains 40961 rows and 33 columns, which is so extensive that it inevitably covers some missing data. Thus, if the data is not processed, the accuracy of the simulation results will be affected.

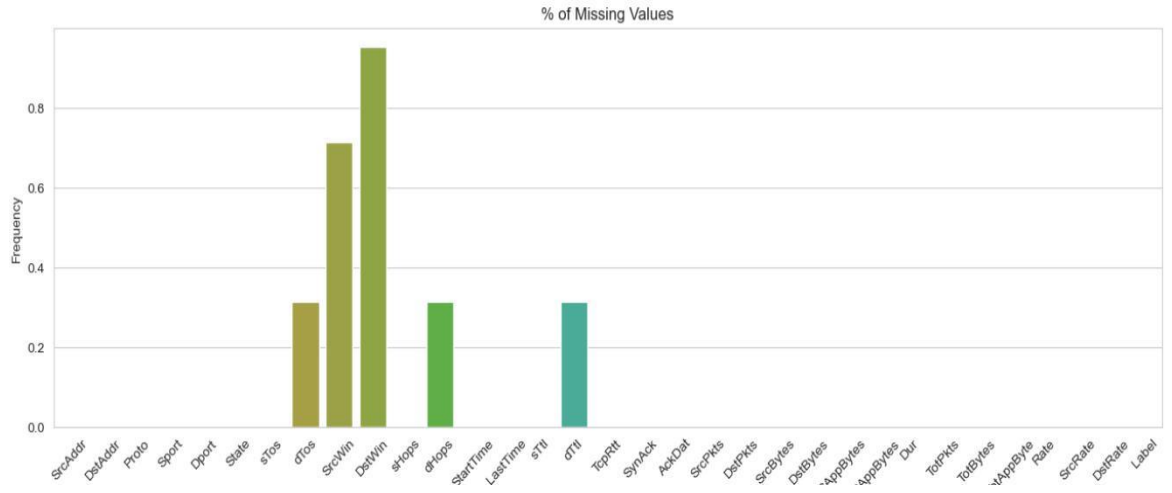

Figure 1. Missing data in the dataset used in DDOS attack detection.

Since this data includes a large-scale capture, which mixes real botnet traffic, normal traffic and background traffic, it is possible to eliminate those missing data. Meanwhile, the most missing part of data accounts for just less than one percent of the total sample size; in other words, the removal of missing data does not significantly show impact on the simulation results. The feature types and proportion of missing data are shown in figure 1.

After removing the missing data, features are going to be processed. It is not only because the data contains both strings like “Source Window” and float numbers like “Transmission Rate”, which are mutually incompatible for the same algorithm in most cases but also because some features are meaningless to the algorithm, such as “Destination Address”. First, some features of the data such as “Proto” and “Source Port” are classified to verify whether the data can be applied to some specific algorithms, for this feature is the dependent variable for the test and the resulting quantity for the simulation. The result shows that it can be used in algorithms like decision trees. Processing features “Start Time” and “Last Time” of the data is the next work, because they are only valid variables after processing. Specifically, their difference will be calculated as a new variable to replace their role in the model. Afterwards, invalid variables “Source Address”, “Destination Address”, “Start Time” and “Last

DOI: 10.54254/2755-2721/18/20230966

Time" are eliminated to ensure that they will not affect the results of model training. So far, the data and features of the DDOS attack detection simulation have been successfully processed.

3.2. SPAM & malware

In terms of the SPAM Email detection simulation, 5572 samples have been collected to verify the relationship between the content of the email and whether it is a SPAM. Figure 2 shows some examples of sample email content. Unlike DDOS attack detection, there is no missing data in this simulation. The main purpose of processing this data is to pick up some key features from sample emails. For statistical purposes, binary is utilized in order to represent whether an email is spam or not. Meanwhile, the CountVectorizer function is used to convert the contents of the mail into feature vectors so that they can be utilized by the algorithms. That is all about how to process data for spam detection simulation, and data processing for malware is similar to it in procedure.

Table 1. Part of the email samples used for detection.

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

4. Algorithms selection

As the features of data needed to be processed in different projects are quite different, the selection of algorithm should be considered separately for each project. As DDOS attack detection contains a large amount of content that cannot be processed mathematically, linear regression cannot be selected in this simulation. Algorithms that can process these types of features, such as logistic regression, decision trees, etc., become better alternatives. After preliminary data processing, it can be found that the accuracy of logistic regression algorithm and naive Bayes algorithm is relatively low, which means they're not good choices for algorithms. As a result, decision tree and random forest are selected, for they all have the accuracy over 99 percent. In the same way, KNN algorithm and decision tree algorithm are also adopted in spam detection and malware detection.

5. Parameters tuning

Although some algorithms with low accuracy have been eliminated in the process of algorithm selection, it is still not guaranteed that the selected algorithm can fit the model, and overfitting phenomenon will not occur. In order to make the model as accurate as possible, parameter adjustment is an essential step in this simulation.

In decision tree algorithm, according to sk-learn tutorial, criterion, max depth, min samples split and min samples leaf are chosen as the parameters to be tuned. Although other parameters can also affect the results of the experiment, they are far less influential than the parameters selected. To begin, the criterion of the model should be determined, though it is not a parameter need to be tuned in most cases. In general, the Gini coefficient is more applicable than the entropy coefficient; nevertheless, whether Gini coefficient is better still needs to be verified for this project. After testing, it can be obtained that the Gini coefficient of the project is 99.40%, and the entropy coefficient is 99.38%. The Gini coefficient proved to be a better criterion for the project. In this and subsequent tests, cross validation will be used to ensure the accuracy of test results and avoid overfitting. Based on this criterion, the most suitable max depth will be tested then.

DOI: 10.54254/2755-2721/18/20230966

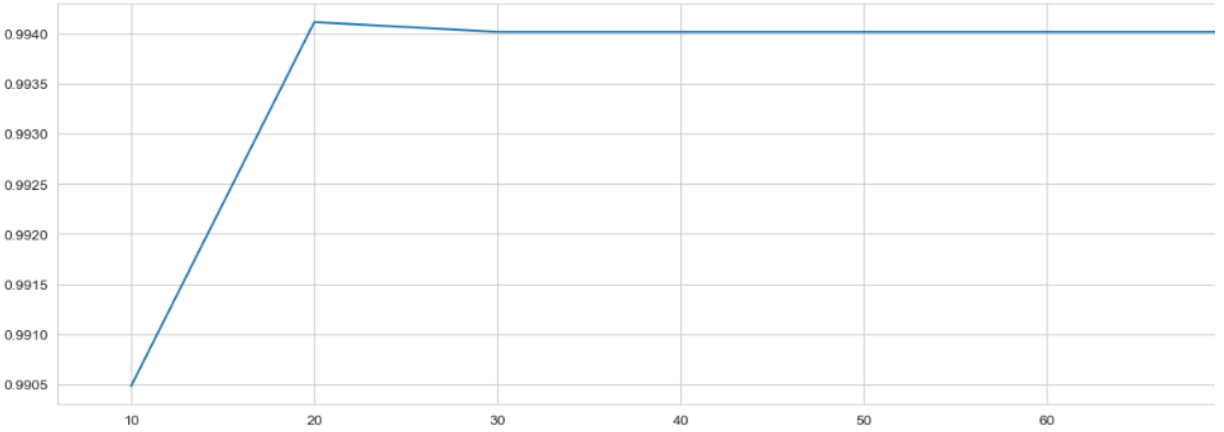

Figure 2. Rough test for optimum max depth.

For the maximum depth, a wide area search is first conducted in order to determine the range in which the optimal value exists. After testing the maximum depth in the range of 1 to 100, it is found that the optimal value exists around 20, just as shown in Figure 3 and Figure 4. Thus, the next job is to search this area precisely. After precise testing on a range of 10 to 25, it can be determined that the optimal value is 20.

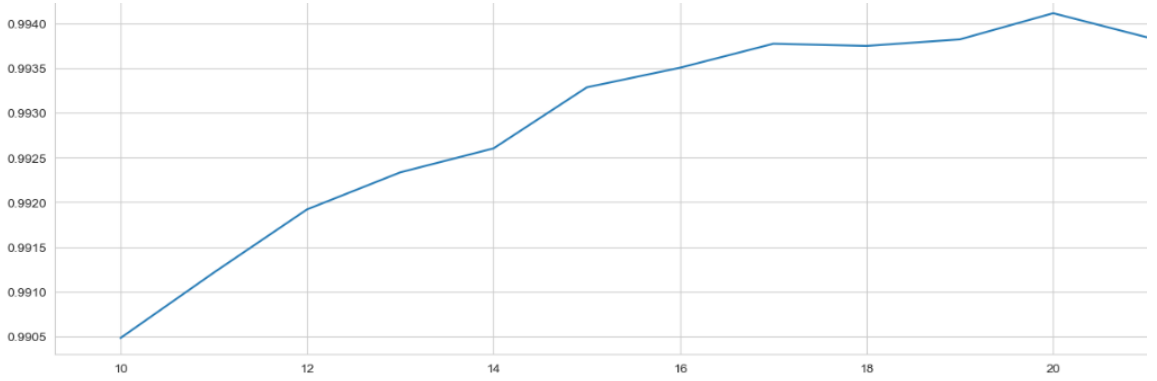

Figure 3. Precise test for optimum max depth.

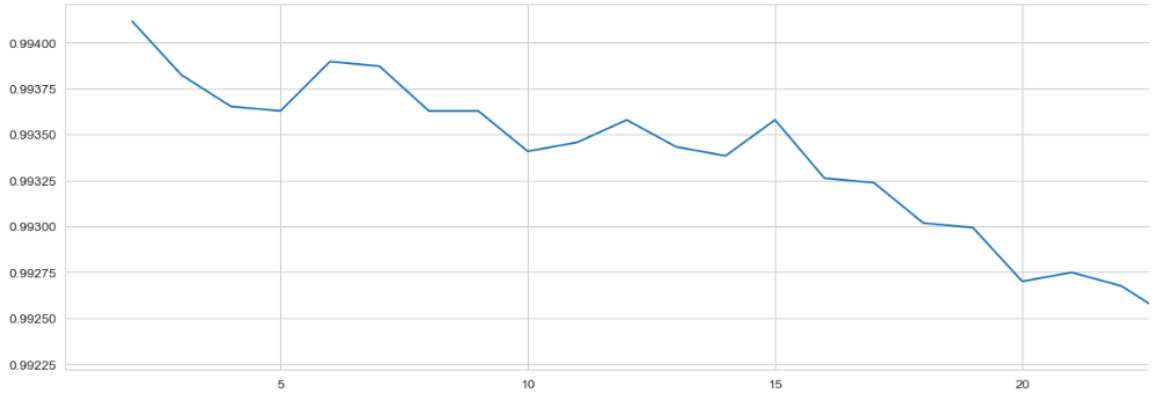

Figure 4. Test for min samples split.

DOI: 10.54254/2755-2721/18/20230966

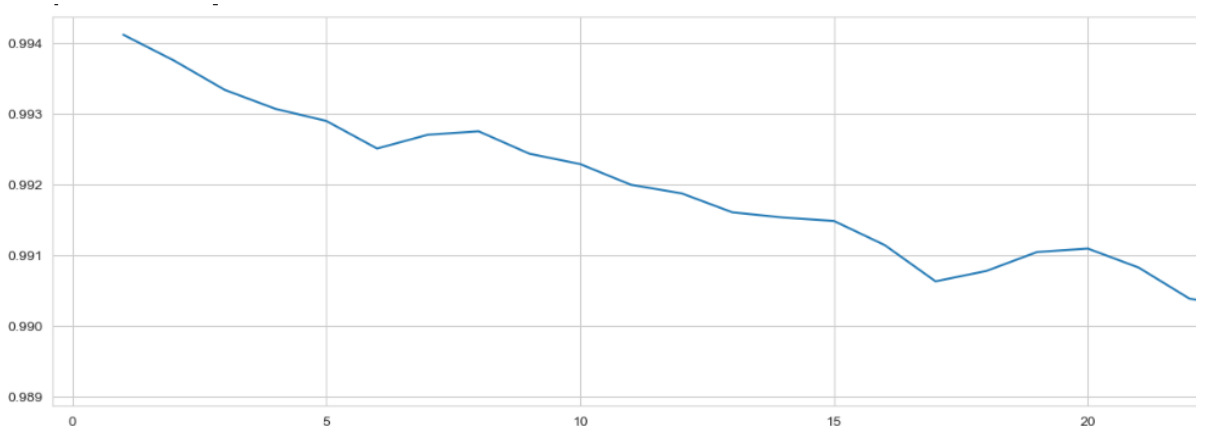

Figure 5. Test for min sample leaf.

Furthermore, the min samples split and the min samples leaf are also tuned. As is shown in Figure 5, the min samples split in the range of 2 to 30 is tested and it is found that the optimal value is 2. Meanwhile, Figure 6 indicates that the min sample leaf is determined to be 1. At this point, all parameters have been tuned. However, this result does not mean that our parameter tuning work is complete. In fact, some parameters are corresponding parameters, and they affect each other; therefore, further parameter tuning is required to ensure the maximum accuracy of the model. After grid search, the parameter combination tested is proved to be the combination with the highest accuracy of the model, which is 99.411%.

In random forest algorithm, the process of parameter tuning is the same as the decision tree algorithm. To be specific, the first task is to test out the most suitable n-estimator. On this basis, maximum depth, min samples split and min sample leaf are then tested out. Specifically, the n-estimators are 76, the max depth is 70, the min samples leaf is 1 and the min samples split is 5. However, a problem arises during grid search. The results of the grid search are not the same as the results of sequential testing, and the accuracy become even lower than the beginning. Maybe it is because only a small range of parameters are selected to do the grid search. Unfortunately, a large range grid search is out of the capacity of personal computers. Grid search in a wider range is crucial for the further research. Currently, the fitting results are still accurate enough as it is over 99.436%. Thus, the model trained with these parameters will not have serious errors. The parameter tuning of other algorithms such as KNN is also carried out according to this process. Since the process is repetitive, it is not described in detail here.

6. Results

In the simulation of DDOS attack detection, the accuracy of the two algorithms is above 99%, which proves that the model can meet the detection requirements. The results of the experiment are shown in figure 7, which shows how DDOS attacks can be detected by different characteristics.

In the SPAM email detection simulation, the accuracy of the two algorithms is about 95%, and the specific results are shown in the figure 8, which is sufficient to prove that the detection function of SPAM can be basically realized. At the same time, through the analysis of the experimental results of the data, it can be found that most of the mistakes are to identify ordinary mail as spam, the opposite situation is almost non-existent. This result also shows that machine learning can be used to ensure the security of Internet of Things systems. Ordinary mail that is misidentified can also be sorted into sandboxes and quarantined for manual processing. In the simulation of malware detection, the accuracy rate is only about 50%, indicating that there is still a great possibility of progress in malware detection

DOI: 10.54254/2755-2721/18/20230966

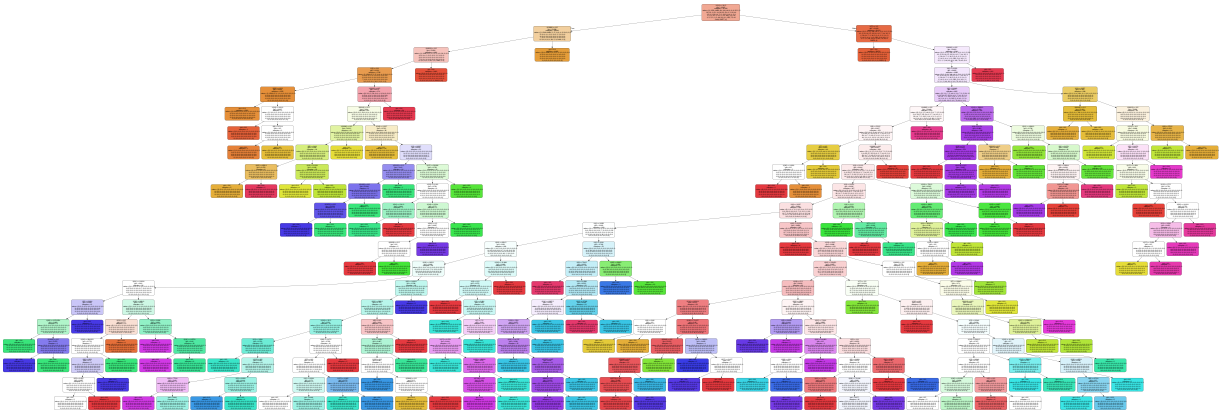

Figure 6. The result of DDOS attack detection with decision tree algorithm.

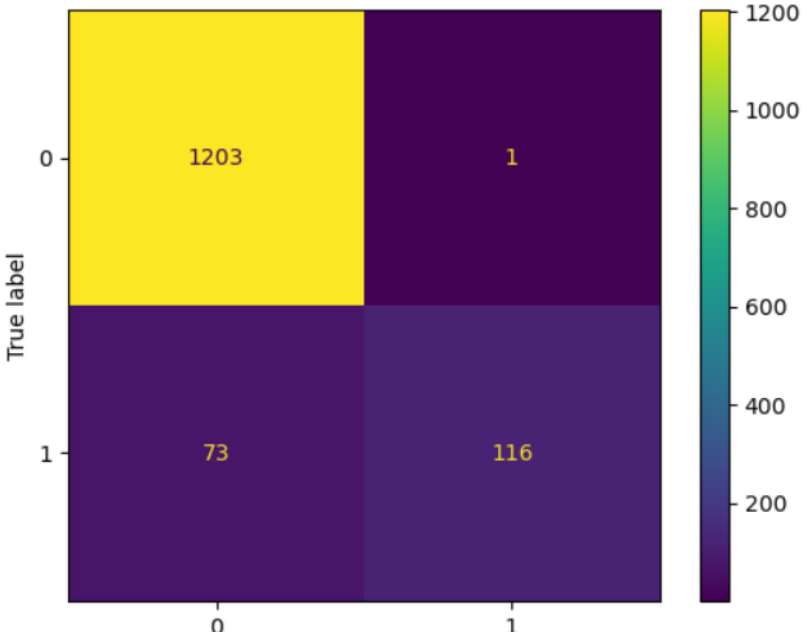

Figure 7. The result of SPAM attack detection with KNN algorithm.

7. Conclusion

After comparing the results of the three detection methods, several conclusions can be drawn. Generally speaking, the more specific and comprehensive the data characteristics used, the more accurate the machine learning model's training results will be. For instance, when detecting Distributed Denial of Service (DDOS) attacks, characteristics such as port and transmission rate can be used to determine if an attack is occurring. However, relying solely on email content for spam detection may result in errors and undermine the accuracy of the results. However, there are some limitations in this study. Firstly, each project relies on a single source of data, which could potentially weaken the reliability of the findings. Using additional data sources could demonstrate the mass applicability of using machine learning for botnet attack detection. Secondly, this study is based on personal computer processing power, which significantly restricts parameter screening, and could lead to issues in the random forest algorithm. The use of server-level computing for data processing would allow for more comprehensive grid searches and ensure greater data accuracy. Finally, data analysis and processing for DDOS attack detection require optimization. In comparison to spam detection, DDOS attack detection necessitates more features, resulting in longer data identification, and analysis time. Over-saturated attacks could still leave IoT systems vulnerable to data detection overloading risks, requiring a focus on optimizing

the data analysis process in future research, ensuring the model keeps up to date with the latest data. Therefore, for future studies, experimental results should be re-evaluated from multiple sources to verify the universality of the conclusions reached in this paper. In DDOS attack detection, the emphasis should be on streamlining the data analysis process to prevent the data feature detection ability of IoT systems from collapsing under botnet saturation attacks while maintaining model accuracy. In spam detection, selecting samples across multiple languages is necessary to validate the universality of detection capabilities across different languages. For malware detection, identifying more representative data characteristics, such as file type, is an essential research direction.

References

[1] J. Liu, S. Liu, and S. Zhang, "Detection of IoT Botnet Based on Deep Learning," in *2019 Chinese Control Conference (CCC)*, Jul. 2019. doi: https://doi.org/10.23919/chicc.2019.8866088.

[2] B. Bojarajulu, S. Tanwar, and T. P. Singh, "Intelligent IoT-BOTNET Attack Detection Model with optimized Hybrid Classification Model," *Computers & Security*, vol. 126, no. 103064, p. 103064, Dec. 2022, doi: https://doi.org/10.1016/j.cose.2022.103064.

[3] S. Sriram, R. Vinayakumar, M. Alazab, and S. KP, "Network Flow based IoT Botnet Attack Detection using Deep Learning," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Jul. 2020. doi: https://doi.org/10.1109/infocomwkshps50562.2020.9162668.

[4] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, p. 110, Mar. 2021, doi: https://doi.org/10.11591/ijai.v10.i1.pp110-120.

[5] J. P, J. Shareena, A. Ramdas, and H. A P, "Intrusion Detection System for IOT Botnet Attacks Using Deep Learning," *SN Computer Science*, vol. 2, no. 3, Apr. 2021, doi: https://doi.org/10.1007/s42979-021-00516-9.

[6] R. Vishwakarma and A. K. Jain, "A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019. doi: https://doi.org/10.1109/icoei.2019.8862720.

[7] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet Detection in the Internet of Things using Deep Learning Approaches," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018. doi: https://doi.org/10.1109/ijcnn.2018.8489489.

[8] G. D. Nguyen, Braulio Dumba, Q.-D. Ngo, H. Le, and Tu Dinh Nguyen, "A collaborative approach to early detection of IoT Botnet," *Computers & Electrical Engineering*, vol. 97, no. 107525, pp. 107525–107525, Nov. 2021, doi: https://doi.org/10.1016/j.compeleceng.2021.107525.

[9] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid Deep Learning for Botnet Attack Detection in the Internet of Things Networks," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 1–1, 2020, doi: https://doi.org/10.1109/jiot.2020.3034156.

[10] J. Kim, M. Shim, S. Hong, Y. Shin, and E. Choi, "Intelligent Detection of IoT Botnets Using Machine Learning and Deep Learning," *Applied Sciences*, vol. 10, no. 19, p. 7009, Oct. 2020, doi: https://doi.org/10.3390/app10197009.