http://dx.doi.org/10.12785/src/1571093544

Bashilite and Mirai Malware Attacks Detection on Internet of Things Devices Using Machine Learning

1st Ruuhwan Ruuhwan
School of Computing
Telkom University
Bandung, Indonesia
ruhwan@student.telkomuniversity.ac.id

2nd Rendy Munadi
School of Computing
Telkom University
Bandung, Indonesia
rendymunadi@telkomuniversity.ac.id

3rd Hilal Huda
School of Computing
CAATIS, Tel-U
Bandung, Indonesia
hilalh@teluapp.org

4th Erwin Budi Setiawan
School of Computing
Telkom University
Bandung, Indonesia
erwinbudisetiawan@telkomuniversity.ac.id

Abstract-- The risk of cyber attacks on Internet of Things (IoT) infrastructure is substantial, especially with devices operating on existing network systems, such as those targeted by Bashilite and Mirai malware. Network forensics investigations necessitate the use of machine learning algorithms for effective classification and detection of these malware attacks. In a series of experiments, five algorithms were tested: Naïve Bayes (NB), Random Forest (RF), Decision Tree (DT), Neural Network (NN), and K-Nearest Neighbor (KNN). The findings indicated that the RF algorithm performed the best, with an average accuracy of 93.78%, recall of 86.3%, F1 score of 90.04%, and the highest precision at 95%. Furthermore, the RF algorithm is particularly adept at handling large datasets. This study suggests that the RF algorithm is an excellent choice for classifying and identifying Bashilite and Mirai malware attacks within IoT infrastructure.

Keywords-- Bashilite; Machine Learning; Malware; Mirai;

1. Introduction

The Internet of Things (IoT) is a collective network of devices that are connected to one another and that allows for communication between their device [1]. The Internet is gradually becoming more integrated with electronic gadgets that are in residential settings, industrial automation systems, and smart city infrastructures [1]. However, the security vulnerabilities associated with these technological advancements are also on the rise. The Internet of Things (IoT) is particularly vulnerable, with a wide range of hardware, such as IP cameras, home routers, and smart devices, being prime targets [2]. Distributed Denial of Service (DDoS) attacks are one of the most dangerous types of cyberattacks. In these attacks, hackers exploit multiple Internet-connected devices, known as botnets, to send overwhelming requests to a server or group of servers, effectively flooding them and preventing legitimate users from accessing the service [3]. Malware, a type of malicious software [4], plays a crucial role in these attacks. Notably, the Bashlite and Mirai malware have been among the most dangerous in recent years, even being responsible for the largest recorded DDoS attack [5].

Bashlite is a type of malware designed to attack Internet of Things (IoT) devices with Linux operating systems running MIPS and ARM architecture processors [6]. Bashlite targets IoT devices that are directly linked to the internet without firewalls or protection [7]. Bashlite's common modus operandi involves exploiting existing security vulnerabilities, usually through the insecure Telnet protocol or unprotected Remote Desktop Protocol (RDP) service. Bashlite has the ability to infect devices quickly and automatically after finding vulnerable devices [8]. This botnet can be commanded to carry out mass DDoS attacks against targets specified by the attacker controlling the botnet. Efforts to combat Bashlite include implementing better security practices on IoT devices, such as changing default passwords, disabling unnecessary Telnet services, and regularly updating software to address known vulnerabilities [9]. Mirai is a type of malware or botnet that is notorious for attacking IoT (Internet of Things) devices by exploiting security weaknesses in commonly overlooked IoT devices such as surveillance cameras, routers, and other devices connected to the internet [10]. Mirai has the ability to quickly infect devices using a list of common or weak default passwords. After successfully infecting a device, Mirai will connect the device to a botnet network controlled

by the attacker [11]. Then, this botnet may be used to launch DDoS attacks against specific targets.

Machine Learning plays a critical role in processing and analyzing the complex data generated by IoT systems [12]. This enables to development of methods to detect and identify new types of attacks, especially by detecting anomaly-based intrusions in the Internet of Things network [13]. This capability is essential for supporting forensic processes, enabling quicker and more accurate decision-making [14]. Such a system requires the ability to learn in real-time and detect previously unseen anomalous patterns.

Given that IoT devices often have limited computational resources [15], the chosen machine learning algorithm must be both computationally efficient and capable of operating effectively on devices with constrained computing power [16]. To address these needs, a comparison of various machine learning algorithms—such as Naïve Bayes (NB), Random Forest (RF), Decision Tree (DT), Neural Network (NN), and K-Nearest Neighbor (KNN)—was conducted. The evaluation focused on criteria including detection accuracy, computational efficiency, and the ability to manage complex and diverse data. Extracted Features can be seen in Table 1.

Table 1. Extracted Features

Value	Statistics	Aggregated by	Total Number of Features
Package Size	Mean, Variance	Source IP, Source MCA-IP, Channel, Socket	8
Packet Count	Number	Source IP, Source MCA-IP, Channel, Socket	4
Packet Jitter	Mean, Variance, Number	Channel	3
Package Size	Magnitude, Radius,Covariance, Correlation Coefficient	Channel Socket	8

In this study, we utilized a public dataset from the UCI Repository, specifically titled IoT Botnet Detection using the N-BaIoT dataset. By using port mirroring on switches that send typical organizational traffic, this dataset was collected from raw network traffic data in pcap format [17].

A summary of the contributions of this research is as follows:

- Detect bashlite (also known gafgyt) and mirai attacks on IoT networks using five machine learning algorithms to evaluate the best performance in detecting attacks from two IoT botnets.
- The data uses 11 classes, bashilite (combo, junk, scan, tcp, udp) and mirai (ack, scan, syn, udp, udpplain), by combining 7 IoT devices in the analysis process with five different methods to determine the best performance in detecting attacks from these 2 IoT botnets instantly.

This paper consists of several discussion sections as follows: Section 2 discusses the research method, Section 3 discusses the results of the experiment, and Section 4 is the conclusion of this paper.

2. Research Methods

The DT algorithm is used to analyze cyber attacks that occur. The accuracy obtained is 97.29% [18]. However, after the Pi camera device was added, there was a decrease in accuracy to 96.01% [18]. RF is used to help identify Mirai attacks so that the identification process can be carried out quickly and accurately. Each decision tree will categorize network flows that are considered dangerous or legitimate flows. The accuracy obtained is 99.75% [19]. The application of NB is used to identify and classify IoT device attacks with the DoS type. The classification results have an accuracy of 64.02% [20]. As the frequency of Android devices rises, malware continuously generates new viruses, endangering the security of the central system and user privacy. To predict Android malware attacks on IoT devices, the KNN algorithm is applied. The results that obtained from the experiment, KNN has an accuracy of 93% [21].

In this study, (Figure 1) is proposed as a research flow stage, there are 3 main stages, namely: Lab Simulation, Data Preprocessing & Analysis and Application of Machine Learning Algorithms.

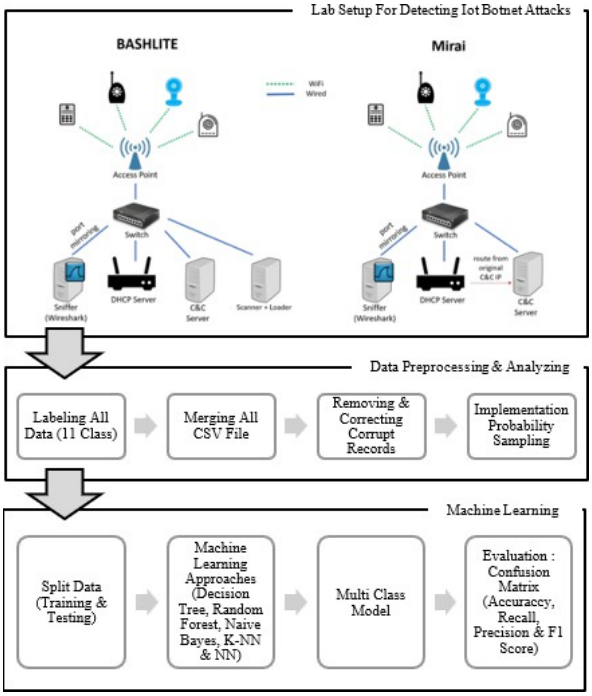

Fig 1. Research Framework.

2.1. Simulation Process

The attack simulation using Mirai and Bashlite within the context of IoT forensics aims to achieve specific objectives, particularly in supporting the investigation, analysis, and in-depth understanding of attacks on IoT devices [22].

Mirai and Bashlite attack simulations allow for the development and testing of new methods to detect and

identify signs of botnet attacks on IoT devices. This includes the development of anomaly detection algorithms or network traffic pattern analysis that can be used to recognize suspicious behavior from infected devices [23].

Attack simulations can be used to validate forensic tools and techniques used in IoT security investigations. This includes tools for obtaining digital evidence from IoT devices, analyzing discovered malware, and reconstructing attack paths that occurred [24].

Through attack simulation, forensics teams can reconstruct the events of a botnet attack from start to finish, including how the malware entered the IoT device, the activities performed by the malware, and the impact it had on the infected system or network.

2.2. Data Preprocessing & Analysis Process

Data preprocessing and analysis are very important stages in the application of machine learning to ensure that the data used is of high quality and ready to be processed by the model. Good data preprocessing and analysis are very important because the quality and relevance of the data directly affect the performance and accuracy of the resulting machine learning model. This stage ensures that the data used is not only clean and ready to be processed, but also provides valuable insights for data-based decision making [25]. The stages can be seen in Figure 2.

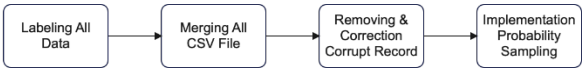

Fig 2. Data Preprocessing & Analysis Process

1. Labeling All Data: Assigning attributes as labels or targets within the dataset, resulting in a total of 11 classes.
2. Merging All CSV Files: Combining all CSV files corresponding to the 11 available labels into a single dataset.
3. Removing and Correcting Corrupt Records: Identifying and either removing or correcting data entries that are inaccurate or corrupted.
4. Implementing Probability Sampling: Applying a data sampling method to prepare the data for further processing.

2.2 Machine Learning Process

The application of machine learning to IoT forensics is one of the important approaches in addressing the increasingly complex security challenges in the IoT environment. Machine learning can be used to detect anomalies in network traffic data or IoT device behavior. This technique helps identify unusual or suspicious activity that may indicate an attack or unauthorized use of the device. The stages can be seen in Figure 3.

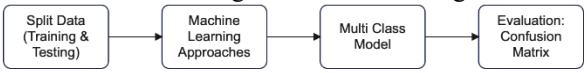

Fig 3. Machine Learning Process

1. Split Data: divides data into 2, namely training data and testing data of 70:30
2. Machine Learning Approaches: applying 5 machine learning algorithms for comparison, namely the NB, DT, RF, NN and KNN algorithms.

A. Naïve Bayes (NB)

The NB algorithm used for malware detection refers to the application of the Naïve Bayes classification method in identifying whether a file or program is malware or not based on features or attributes related to the behavior or characteristics of the malware [26]. The advantages are:

a) It is relatively simple and fast to train the model and make predictions for malware detection because speed in classifying samples is very important.
b) Efficient in handling large features because malware analysis takes into account various file attributes and behaviors.
c) The predictions are easy to interpret because they use probability to determine the classes.
d) Can work well in performing malware analysis to identify suspicious patterns and behavior.

The disadvantages of the NB algorithm are as follows:

a) Always assume that all features in the dataset are conditionally independent with respect to the target class.
b) Unable to handle non-linear relationships between existing features, malware detection can be problematic because relevant patterns and features may not always satisfy this assumption.
c) If there are missing values in the data, special techniques are required to handle such cases without significantly reducing performance.

NB can be used for malware detection due to its speed. However, its independence assumption and its limitations in handling complex relationships between features can be limitations in cases where the relationships between features are critical.

B. Decision Tree (DT)

The DT algorithm is used to perform classification and regression so that it can create a predictive model in the form of a decision tree structure [27]. The advantages are as follows:

a) It is easy for humans to understand and interpret because its structure is similar to a rule-based thought process (if-then-else).
b) Capable of handling non-linear data between complex malware behavioral features without requiring special data normalization or transformation.
c) Capable of handling mixed features such as numeric and categorical features.
d) Efficient in data processing because it only requires repeated separation of data based on feature values.
e) Able to handle missing values in datasets with missing values without requiring complex imputation.

The disadvantages of the DT algorithm are as follows:

a) Too much detail in studying the training data can cause overvitalization.
b) Small changes to the training data can produce significant differences, so can cause inconsistency in generalization
c) Unstable to data changes resulting in differences in results that can impact the interpretation and reliability of the results.
d) Inefficient in representing very complex relationships between existing features.
e) Not effective for dealing with regression issues in the context of malware detection.

The DT algorithm can be used for malware detection due to its high interpretability and ability to handle non-linear relationships. However, overfitting and inconsistency of results can be significant problems, which require special attention when designing and using such models for malware detection.

C. Random Forrest (RF)

The RF algorithm is often used for malware detection to identify whether a file or digital activity is malware or not [18]. The advantages are as follows:

a) Provides accurate prediction results because it combines predictions from many different decision trees.
b) Can reduce the risk of overfitting compared to a single decision tree.
c) Can handle datasets with many features and poorly structured data.
d) Relatively stable to changes in training data and not too sensitive to small data changes.
e) It is easy to implement and does not require a lot of parameter tuning like some other machine learning techniques.

The disadvantages of the RF algorithm are as follows:

a) The training and prediction process can be slower and require more computing resources.
b) It is not always easy to interpret intuitively, especially when there are many decision trees used.
c) Tends to provide predictions that favor the majority class.
d) If the selection of features or attributes used is not optimal, performance can be negatively affected.
e) Vulnerable to noise or irrelevant data if not managed properly in the feature selection process.

The main advantages of Random Forest in the context of malware detection include its ability to overcome overfitting (modeling the training data too well), as well as its ability to handle various types of complex and unstructured features or attributes.

D. Neural Networks (NN)

The NN algorithm is one approach in creating models for malware detection[28]. Its advantages are as follows:

a) Capable of learning very complex representations from data, including patterns that are difficult to detect by conventional malware detection methods.
b) It can be used to process various types of features relevant in malware detection, such as file metadata, execution behavior, or network traffic patterns.
c) Can be configured to adapt to changes in the properties used by malware.
d) Can produce accurate predictions in detecting malware.

The disadvantages of the NN algorithm are as follows:

a) Requires large and varied amounts of training data to learn well.
b) Time consuming computational process
c) It is difficult to clearly understand why a decision was made, which can make it difficult to analyze and handle false positive or missed false negative detection results.
d) Vulnerable to attacks and manipulation, such as adversarial attacks, where small changes in the input can cause large changes in the predicted output.

The use of NN in malware detection offers great potential to improve the ability of detection systems to deal with increasingly complex and diverse threats. However, its use requires a balance between the need for large and representative data with the challenges of interpreting and managing model complexity.

E. K-Nearest Neighbourhood (KNN)

The KNN algorithm is used in the context of classification to identify whether a file or activity is malware or not based on the attributes possessed by the data [20]. The advantages are as follows:

a) The simplest and easiest to understand classification algorithm.
b) It does not require an extensive training process like other machine learning algorithms.
c) It is generally effective in detecting anomalies or outliers, as it relies on calculating distances to the nearest neighbors.
d) It supports multi-class classification without the need for additional customization.
e) It does not assume a specific distribution of the data or a linear relationship between features and labels, making it effective in handling data with non-linear patterns.

However, the K-NN algorithm has some disadvantages:

a) It can be time-consuming to calculate the distance from a new data point to all training data points, especially in large datasets.
b) Its performance can suffer if the dataset contains many irrelevant features or noise.
c) The algorithm's effectiveness is highly dependent on the proper selection of the K parameter. A too-small K value can make the model vulnerable to noise, while a too-large K value can make the model less sensitive to important patterns. Too

large K value can cause the model to be too general and less sensitive.

d) Tends to be unable to understand complex relationships between features and labels in complex data.

K-NN has simple and easy to understand properties and can be effective in detecting anomalies, its main weaknesses lie in its limitations in handling large and complex data, sensitivity to irrelevant features, and dependence on the selection of the right K parameter.

1. Multi Class Model: processes existing data using a predetermined algorithm.
2. Evaluation: conducting an evaluation to measure the performance of each algorithm.

By using machine learning algorithms, several types of metrics can be generated, such as accuracy, precision, recall, and F1 score. Accuracy is the ratio of the total number of correct classifications, including true positives (TP) and true negatives (TN), predicted by the algorithm to the total number of samples, including true positives (TP), true negatives (TN), and false positives (FP). This can be represented mathematically as follows

$$Accuracy = \frac{TN+TP}{FN+TN+TP+FP} (1)$$

.

Precision represents the ratio of TP to the total TP and FP. This metric shows the number of positive instances correctly classified out of all positive instances.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Sensitivity, also known as Recall, is the ratio that shows how many positive samples are correctly identified from all the actual positive samples in the dataset. It measures the model's ability to find all relevant instances.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1 score is a metric that combines precision and recall to provide a single performance value for the model. It can be calculated as the harmonic mean of both metrics, offering a balanced evaluation when both metrics are important.

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (4)$$

3. Results and Discussions

The Random Forest Confusion Matrix Model provides a focused evaluation of the accuracy performance for the N-BaIoT dataset (Class 11). As illustrated in Figure 4, it offers a centralized assessment of the accuracy achieved by five machine learning algorithms above. The dataset is divided into a 70% training set and a 30% testing set. Centralized Model Performance Using Multiclass can be seen in Table 2.

Table 2. Centralized Model Performance Using Multiclass (11 classes)

Algorithm	Performance					
	Class Error	Precision	Recall	Time (Sec)	Accuracy	F1-Score
Naive Bayes	44%	0.523	0.516	2,709	55.63%	0.520
Random Fores	6%	0.950	0.863	11,747,720	93.78%	0.904
Decision Tree	28%	0.629	0.619	240,896	72.35%	0.624
Neural Network	20%	0.853	0.836	983,873	86.32%	0.844
K-Nearest Neighbor	15%	0.874	0.858	24,562,973	85.11%	0.866

Figure 4 illustrates the performance comparison of the machine learning algorithms. The confusion matrix provides an overview of the classifiers' performance on the test dataset by comparing predicted values against actual true values. The terms used in the confusion matrix include FP, TP, FN, and TN.

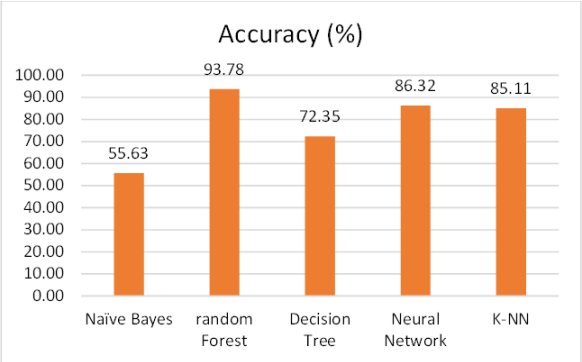

Fig. 4. All Models Accuracy

Classification error refers to the error rate produced by the model, with a lower error rate indicating better model performance. The RF algorithm achieved the lowest classification error at 6%, while the Naive Bayes algorithm had the highest error at 44%. The Neural Network (NN) yielded an error rate of 13%, the Decision Tree 28%, and K-NN 15%.

The F1-Score ranges from 0 to 1, where 1 represents the best possible performance. A high F1-Score indicates that the classification model has strong precision and recall. The Random Forest algorithm achieved the highest F1-Score at 0.904, while the Naive Bayes algorithm had the lowest at 0.523. The Decision Tree obtained an F1-Score of 0.624, Neural Network 0.844, and K-NN 0.866.

In terms of accuracy, the Random Forest classifier performed the best, achieving 93.78%. The lowest accuracy was observed with the Naive Bayes classifier at 55.63%. The Decision Tree classifier reached 72.35% accuracy, the Neural Network 86.32%, and the K-NN classifier 85.11%.

.

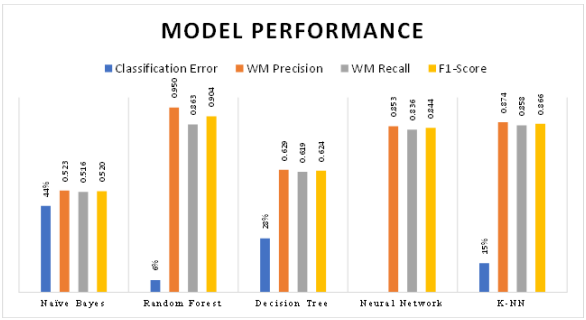

Fig. 5. Centralized Model Performance Using Multiclass (11 classes)

The longest training time required to create a model was obtained with the K-NN algorithm with a time of 24,562,973 seconds, Random Forest with a time of 11,747,720 seconds, Decision Tree with a time of 240,896 seconds and Naive Bayes with a time of 2,709 seconds.

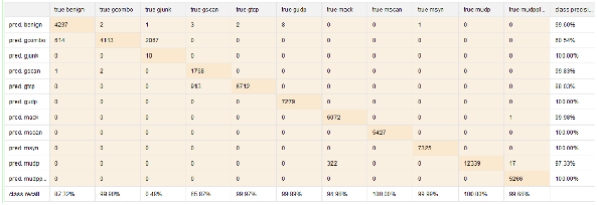

Fig. 6. Confusion Matrix Of Random Forest Using Multiclass (11 classes)

4. Conclusions

The conclusion of this study is that the random forest algorithm is the best model performance in identifying attacks from 7 IoT devices against 2 botnets in the network, where the lowest classification error value is 6%, the F1-Score value closest to 1 is 0.904 and the highest accuracy is 93.78%. However, the time required to train the data tends to be longer than Naive Bayes and Decision Tree.

In this study, the implementation of the five algorithms was applied one by one. For future research, we would like to try combining several machine learning algorithms to improve performance in detecting malware.

References

[1] D. Bastos, M. Shackleton, and F. El-Moussa, "Internet of Things: A Survey of Technologies and Security Risks in Smart Home and City Environments," in Living in the Internet of Things: Cybersecurity of the IoT - 2018, London, UK: Institution of Engineering and Technology, 2018, p. 30 (7 pp.)-30 (7 pp.). doi: 10.1049/cp.2018.0030.

[2] AN Irfan, A. Ariffin, MN Mahrin, and S. Anuar, "A Malware Detection Framework Based on Forensic and Unsupervised Machine Learning Methodologies," in Proceedings of the 2020 9th International Conference on Software and Computer Applications, Langkawi Malaysia: ACM, Feb . 2020, pp. 194–200. doi: 10.1145/3384544.3384556.

[3] K. Angrishi, "Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets." arXiv, Feb. 13, 2017. Accessed: Jul. 01, 2024. [Online]. Available: http://arxiv.org/abs/1702.03681

[4] TP Setia, AP Aldya, and N. Widiyasono, "Reverse Engineering for Remote Access Trojan Malware Analysis," JEPIN, vol. 5, no. 1, p. 40, Apr. 2019, doi: 10.26418/jp.v5i1.28214.

[5] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation," Security and Communication Networks, vol. 2018, pp. 1–30, 2018, doi: 10.1155/2018/7178164.

[6] A. Marzano et al., "The Evolution of Bashlite and Mirai IoT Botnets," in 2018 IEEE Symposium on Computers and Communications (ISCC), Natal: IEEE, June. 2018, pp. 00813–00818. doi: 10.1109/ISCC.2018.8538636.

[7] R. Kawasoe, C. Han, R. Isawa, T. Takahashi, and J. Takeuchi, "Investigating behavioral differences between IoT malware via function call sequence graphs," in Proceedings of the 36th Annual ACM Symposium on Applied Computing, Virtual Event Republic of Korea: ACM, Mar. 2021, pp. 1674–1682. doi: 10.1145/3412841.3442041.

[8] Đ. D. Jovanović and PV Vuletić, "Analysis and Characterization of IoT Malware Command and Control Communication," Telfor Journal, vol. 12, no. 2.

[9] M. Shobana, "International Journal of Scientific Research in Computer Science, Engineering and Information Technology," 2018.

[10] A. Rahmatulloh, G. Muhammad Ramadhan, I. Darmawan, N. Widiyasono, and D. Pramesti, "Identification of Mirai Botnet in IoT Environment through Denial-of-Service Attacks for Early Warning Systems," JOIV : Int. J. Inform. Visualization, vol. 6, no. 3, p. 623, Sept. 2022, doi: 10.30630/joiv.6.3.1262.

[11] R. Doshi, N. Apthorpe, and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," in 2018 IEEE Security and *Privacy Workshops (SPW)*, San Francisco, CA: IEEE, May 2018, pp. 29–35. doi: 10.1109/SPW.2018.00013.

[12] S. Strecker, R. Dave, N. Siddiqui, and N. Seliya, "A Modern Analysis of Aging Machine Learning Based IoT Cybersecurity Methods," JCSA, vol. 9, no. 1, pp. 16–22, Oct. 2021, doi: 10.12691/jcsa-9-1-2.

[13] AA Megantara and T. Ahmad, "A hybrid machine learning method for increasing the performance of network intrusion detection systems," J Big Data, vol. 8, no. 1, p. 142, Dec. 2021, doi: 10.1186/s40537-021-00531-w.

[14] MS Mazhar et al., "Forensic analysis on internet of things (iot) devices using machine-to-machine (m2m) framework," Electronics, vol. 11, no. 7, p. 1126,Apr.2022, him: 10.3390/electronics11071126.

[15] K. Shaukat, TM Alam, IA Hameed, WA Khan, N. Abbas, and S. Luo, "A Review on Security Challenges in Internet of Things (IoT)," in 2021 26th International Conference on Automation and Computing (ICAC), Portsmouth, United Kingdom: IEEE, Sep. 2021, pp. 1–6. doi: 10.23919/ICAC50006.2021.9594183.

[16] N. Scheidt and M. Adda, "Identification of IoT Devices for Forensic Investigation," in 2020 IEEE 10th International Conference on Intelligent Systems (IS), Varna, Bulgaria: IEEE, Aug. 2020,pp. 165–170. doi: 10.1109/IS48319.2020.9200150.

[17] Y. Meidan et al., "N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders," IEEE Pervasive Comput., vol. 17, no.

3, pp. 12–22, Jul. 2018, doi: 10.1109/MPRV.2018.03367731.

[18] M Zeeshan Arshad et al., “Digital Forensics Analysis of IoT Nodes using Machine Learning,” JCBI, vol. 4, no. 01, Dec. 2022, doi: 10.56979/401/2022/107.

[19] N. Widiyasono, IA Dwi Giriantari, M. Sudarma, and L. Linawati, "Detection of Mirai Malware Attacks in IoT Environments Using Random Forest Algorithms," TEM Journal, pp. 1209–1219, Aug. 2021, doi: 10.18421/TEM103-27.

[20] FF Setiadi, MWA Kesiman, and KYE Aryanto, "Detection of dos attacks using naive Bayes method based on internet of things (iot)," J. Phys.: Conf. Ser., vol. 1810, no. 1, p. 012013, March. 2021, doi: 10.1088/1742-6596/1810/1/012013.

[21] H. Babbar, S. Rani, DK Sah, S.A. AlQahtani, and A. Kashif Bashir, “Detection of Android Malware in the Internet of Things through the K- Nearest Neighbor Algorithm,” Sensors, vol. 23, no. 16, p. 7256, Aug. 2023, doi: 10.3390/s23167256.

[22] E. Yusuf Güven and Z. Gürkaş-Aydin, “Mirai Botnet Attack Detection in Low-Scale Network Traffic,” Intelligent Automation & Soft *Computing*, vol. 37, no. 1, pp. 419–437, 2023, doi: 10.32604/iasc.2023.038043.

[23] S.-Y. Hwang and J.-N. Kim, “A Malware Distribution Simulator for the Verification of Network Threat Prevention Tools,” Sensors, vol. 21, no. 21, p. 6983, Oct. 2021, doi: 10.3390/s21216983.

[24] F. Khan et al., “Development of a Model for Spoofing Attacks in the Internet of Things,” Mathematics, vol. 10, no. 19, p. 3686, Oct. 2022, doi: 10.3390/math10193686.

[25] S. Riaz et al., “Malware Detection in Internet of Things (IoT) Devices Using Deep Learning,” Sensors, vol. 22, no. 23, p. 9305, Nov. 2022, doi: 10.3390/s22239305.

[26] Department of Computer Engineering, Istanbul University-Cerrahpasa, Istanbul, Turkey et al., “Multiple Classification of Cyber Attacks Using Machine Learning,” Electrica, vol. 22, no. 2, pp. 313–320, Jun.2022, him: 0.54614/electrica.2022.22031.

[27] A. Yeboah-Ofori, “Classification of Malware Attacks Using Machine Learning In Decision Trees,” 2020.

[28] R. Nagaraju, JT Pentang, S. Abdufattokhov, R. F. CosioBorda, N. Mageswari, and G. Uganya, “Attack prevention in IoT through hybrid optimization mechanism and deep learning framework,” Measurement: Sensors, vol. 24, p. 100431, Dec. 2022, him: 10.1016/j.measen.2022.100431.