International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Classification of Network Traffic Data Mirai Malware Attacks on Internet of Things Devices Using the K-Nearest Neighbor Method

Cecep Suprayogi[1], M Akbar Marwan[2]
[1,2]Master of Electrical Engineering, University of Gunadarma, Indonesia
Email address: [1]cecepsuprayogi@gmail.com, [2]akbar@staff.gunadarma.ac.id

Abstract**—** *Security on the Internet of Things (IoT) devices is still a concern for all parties, especially the problem of Mirai-type Malware attacks. Mirai is one of the most dangerous malware in recent years. Mirai is a botnet designed to penetrate IoT devices with weak security systems. This study detected Mirai malware using a machine learning method with the K-Nearest Neighbour (KNN) algorithm based on distance calculations using the Euclidean Distance method, which can model the classification of network traffic data for Mirai malware attacks through Confusion Matrix testing of objects tested based on learning data that is closest to the thing. The total data in the dataset of 145868 consists of Benign data of 24774, Mirai ACK data of 24527, Mirai Scan data of 23691, Mirai SYN data of 24515, Mirai UDP data of 23766, and Mirai UDP Plain data of 24595. The K parameter value used in the KNN algorithm is K, as much as 5, the percentage of training data is 70%, and the amount of testing data is 30%. From the testing data, 43581 data were predicted correctly, and 178 data were predicted incorrectly, with the calculation of the number of K as many as 5. This study successfully tested the confusion matrix on the classification model using the KNN algorithm in detecting malware attacks in the attack of IoT device architecture. The results of the tests carried out have a relatively high accuracy of 99.59%.*

Keywords**—** *Malware, Mirai, K-Nearest Neighbour, Classification, IoT.*

I. INTRODUCTION

Advances in information and the development of an increasingly growing virtual world have led to many statistical reports and data that must be maintained because there are various ways hackers obtain personal information and data. The vulnerability of a computer network is also due to the increasing prevalence of knowledge about system hacking. Therefore various hackers who take or steal information through one of the attacks that can be carried out are malware attacks.

Malware is software created for a specific purpose by looking for system security holes. Malware can hurt computers and users because attackers can steal someone's personal information or data. In addition, attackers create malware to damage or break into an operating system through scripts inserted by attackers in secret.

This malware has various types, such as backdoors, botnets, downloaders, launchers, and the latest style of malware called Mirai. Mirai is one of the most dangerous malware in recent years. This botnet-type malware attacks IoT devices by infecting a group of computers and other devices connected to the internet and forcing infected machines to attack the system in a coordinated manner.

Various attack studies have been successfully developed using a method implementation, and this study discusses reducing the spread of malware on IoT devices. Vulnerable IoT devices lure attackers to exploit and turn into botnet-like pools used to perform DDoS attacks. In this study, exploitation techniques and whitelisting-based solutions are proposed to prevent IoT botnets from spreading, and this experiment illustrates the success of Mirai malware blocking [1].

Another study also discussed the analysis of IoT-based network traffic botnets to detect honeynet data using classification methods. Botnet detection is carried out by spreading honeynets by providing activity logs from intrusion attempts and dumping network traffic in the form of packet capture. Botnet detection uses network flow and classification techniques to find the presence of botnets in the network by analyzing network traffic and finding features that significantly affect botnet traffic. The data set obtained from honeynet is used to detect botnets using machine learning classification techniques [2].

Other studies compare several Machine Learning (ML) methods such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), Random Forest (RF), Artificial Neural Network (ANN), and Logistic Regression (LR) which can be implemented into the Intrusion Detection System (IDS). In the case of a Distributed Denial-of-Service (DDoS) HTTP attack, the RF accuracy is 99%. In addition, other simulation result-based precision, recalls, F1 scores and log loss metrics revealed that RF outperformed all attack types in binary classification. However, in the multi-class classification, KNN outperforms other ML algorithms with an accuracy of 99%, which is 4% higher than RF [3].

Based on related problems and research, it can be developed on the classification of network traffic data for malware attacks on IoT devices using the K-Nearest Neighbour (KNN) method. The use of KNN because it has the advantage of being able to classify unknown network traffic data with training data and test data. KNN can process a mathematical basis to evaluate the value of these criteria into classification information. This method can accurately classify the data by selecting in advance the K values of the nearest neighbours exactly. KNN can also sort out the data collection

Cecep Suprayogi and M Akbar Marwan, "Classification of Network Traffic Data Mirai Malware Attacks on Internet of Things Devices Using the K-Nearest Neighbor Method," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 4, pp. 39-43, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

of the Mirai malware attack network traffic used in this study, namely Benign data, Mirai ACK data, Mirai Scan data, Mirai SYN data, Mirai UDP data, Mirai UDP Plain data in IoT device architecture attacks.

II. Literature Review

A. Machine Learning

Machine learning is a series of techniques that can assist in handling and predicting extensive data by presenting the data with learning algorithms [4]. In machine learning, there are scenarios such as [5]:

- Using Supervised Learning scenarios, learning uses labelled learning data input. After that, make predictions from the data that has been labelled.
- Using the Unsupervised Learning scenario, learning uses learning data input that is not labelled. After that, try to group the data based on the characteristics encountered.
- Use of Reinforcement Learning scenarios, the learning and test phases are mixed. To collect learner information actively by interacting with the environment to get a reply to any action from the learner.

B. Internet of Things (IoT)

IoT is a concept in which an object or objects are embedded with technologies such as sensors and software to communicate, control, connect, and exchange data through other devices as long as they are still connected to the internet. Threats to IoT target a wide range of hardware, including IP cameras, home routers, and bright devices. This threat generally affects Linux-based systems. Infected device service using Censys Internet scan revealed that most of its devices were routers and cameras. Mirai actively de-identifies, explaining that it cannot identify most devices [6].

C. K-Nearest Neighbor (KNN)

KNN is a method for classifying the object being tested based on the learning data closest to the thing. The KNN algorithm includes supervised learning algorithms, The results of the new query instance (test object) will be classified in class form based on most of the categories in the KNN. The classification result class is the class that appears the most [7]. Near or far, the distance between one object and neighbouring objects can be calculated based on Euclidean Distance. Euclidean Distance is a formula used to calculate the distance of conformity or proximity. The KNN algorithm calculates the x-coordinate and y-coordinate points and uses Equation (1) as the Pythagorean Theorem.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

The Pythagorean theorem is used to find the distance between two points, called the Euclidean distance.

III. Proposed Method

A. Research Stages

The stages in this research are:

- Determining Research Problems and Objectives
  The problem and the purpose of this research are to implement and test the level of accuracy of the performance of the KNN method in the classification of network traffic data in IoT device attacks.
- Dataset Collection
  Dataset obtained from kaggle.com. Where is the data from the N-BaIoT dataset "Detection of IoT Botnet Attacks". The datasets taken are network traffic data without any attacks and network traffic data from Mirai malware attacks on IoT devices. The dataset consists of six classes: Bening data, Mirai ACK data, Mirai Scan data, Mirai SYN data, Mirai UDP data, and Mirai UDP Plain data.

B. KNN Algorithm Design and Additional Methods

The flowchart of the KNN algorithm and additional methods consists of the following stages:

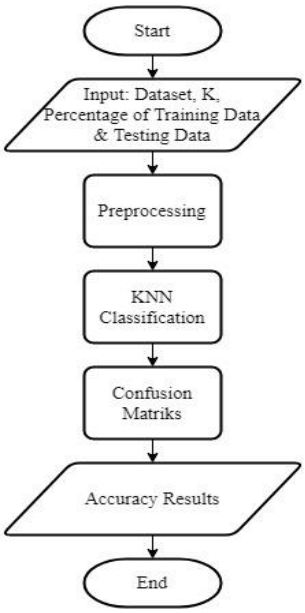

Fig. 1. Flowchart KNN

The explanation for each stage of the KNN Flowchart in Figure 1 is as follows:

- Determine the parameter value in the KNN algorithm, namely K, and the percentage of the amount of training data and testing data. Parameter K is the parameter that takes K data closest to the test data.
- Conducting the Preprocessing stage to divide training data and testing data.
- Classify the KNN algorithm and perform distance calculations using the Euclidean Distance method. Distance calculation to find out the proximity of the test data to the training data. Voting by taking the closest training data with as much test data as K data, then the most dominant class voting of the class on the trained data taken. The voting results will be used to determine the class on the test data.
- Evaluate the classification results using the Confusion Matrix calculation.
- Returns the level of accuracy that the KNN algorithm generates.

C. Modeling

Cecep Suprayogi and M Akbar Marwan, "Classification of Network Traffic Data Mirai Malware Attacks on Internet of Things Devices Using the K-Nearest Neighbor Method," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 4, pp. 39-43, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Modeling is carried out during the Features Selection process, namely selecting all datasets that have been merged according to the criteria for the best features from the Gini index, which will be used for the classification process.

D. Testing and Evaluation

Tests are carried out to measure the classification model formed. In this study using the Confusion Matrix [8]. The confusion matrix is information about the actual classification results that a classification system can predict. The confusion matrix table for 2x2 dimensions is shown in Table 1.

TABLE 1. Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives TP	False Positives FP
Predicted Negative (0)	False Negatives FN	True Negatives TN

The explanations of some of the abbreviations above are as follows:

- TP (True Positive) is the amount of data in the actual class, and the prediction class is also positive.
- FN (False Negative) is the amount of positive data in the actual class, while the prediction class is negative.
- FP (False Positive) is the amount of data in the actual class is negative class while the prediction class is positive.
- TN (True Negative) is the amount of data in the actual class, which is negative, and the prediction class is also harmful.

As for some testing in detail, it will look for Accuracy. Accuracy is a test method based on the closeness between the predicted value and the total actual value. The accuracy formula is as in equation (2).

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

IV. Result and Discussion

The research results consisted of the effects of manual calculations of the KNN algorithm and the consequences of applying measures using the Python programming language.

A. Calculation Results of the KNN Method

1. Input Dataset

The loaded dataset used is data without Mirai malware attacks in the form of Benign data and Mirai malware attack data in the form of Mirai ACK data, MIRAI SCAN data, Mirai SYN data, Mirai UDP data, and Mirai UDP Plain data. From the dataset, several data were taken for samples, as many as 2477 Benign data, 24527 Mirai ACK data, 23691 Mirai SCAN data, 24515 Mirai SYN data, 23766 Mirai UDP data, and 24595 Mirai UDP Plain data. The parameter value used in the manual process of the KNN algorithm is K, as much as 5, and the percentage of training and testing data is 70%: 30%.

2. Preprocessing

The total data in the dataset is 145868. In the manual calculations taken, 10 data at random used for implementing the KNN are shown in Table 2.

TABLE 2. Dataset

No	MI_dir_L5_weight	MI_dir_L5_mean	MI_dir_L5_variance	Type
1	4.060599	101.3228	27.98326	Benign
2	138.8842	192.3111	49443.2	Mirai_ack
3	100.75	478.3924	36654.35	Mirai_ack
4	120.7283	440.9571	47635.93	Mirai_ack
5	40.58121	60.00017	0.008175	Mirai_scan
6	123.3739	60	0	Mirai_scan
7	11.52137	60.05978	2.865629	Mirai_scan
8	104.106	72.99481	21.30818	Mirai_syn
9	1.999941	102	0	Mirai_syn
10	216.9756	66.34812	48.58011	Mirai_syn

Preprocessing is carried out to divide training data and testing data. Here is a calculation to determine the amount of training data.

Data Training=Round(Percentage of Number of Training Data * Number of Data)
Data Training=Round (70% * 10)
Data Training=Round(70/100 * 10)
Data Training=Round (7)
Data Training=7

Here is a calculation to determine the amount of testing data.

Data Testing=Round (Percentage of Number of Testing Data * Number of Data)
Data Testing=Round (30% * 10)
Data Testing=Round(30/100 * 10)
Data Testing=Round (3)
Data Testing=3

TABLE 3. Dataset Training

No	MI_dir_L5_weight	MI_dir_L5_mean	MI_dir_L5_variance	Type
1	4.060599	101.3228	27.98326	Benign
2	138.8842	192.3111	49443.2	Mirai_ack
3	100.75	478.3924	36654.35	Mirai_ack
4	120.7283	440.9571	47635.93	Mirai_ack
5	40.58121	60.00017	0.008175	Mirai_scan
6	123.3739	60	0	Mirai_scan
7	11.52137	60.05978	2.865629	Mirai_scan

TABLE 4. Dataset Testing

No	MI_dir_L5_weight	MI_dir_L5_mean	MI_dir_L5_variance	Type
8	104.106	72.99481	21.30818	Mirai_syn
9	1.999941	102	0	Mirai_syn
10	216.9756	66.34812	48.58011	Mirai_syn

The amount of data used as training data is 7 which is shown in Table 3, while the testing data is 3, which is shown in Table 4.

3. KNN

The distance between the test data and the training data is calculated using the Euclidean Distance method. Here is the distance calculation for the first row and second column based on equation (1).

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d(x_9,x_1) = \sqrt{\left(x_{(8,1)} - x_{(1,1)}\right)^2 + \left(x_{(8,2)} - x_{(1,2)}\right)^2 + \left(x_{(8,3)} - x_{(1,3)}\right)^2}$$

$$d(x_9,x_1) = \sqrt{\begin{matrix}(104.106 - 4.060599)^2 + (72.99481 - 101.3228)^2 \\ +(21.30818 - 27.98326)^2\end{matrix}}$$

$$d(x_9,x_1) = \sqrt{(100.045401)^2 + (-28.32799)^2 + (-6.67508)^2}$$

$$d(x_9,x_1) = \sqrt{10009.08226 + 802.4687853 + 44.55669301}$$

Cecep Suprayogi and M Akbar Marwan, "Classification of Network Traffic Data Mirai Malware Attacks on Internet of Things Devices Using the K-Nearest Neighbor Method," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 4, pp. 39-43, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

$d(x_2,x_1) = \sqrt{10856.10774}$
$d(x_2,x_1) = 104.1926472$

Distance calculation is carried out by entering the value of the testing data for the test calculated by equation (1). The calculation of the 8th testing data can be shown in Table 5, The calculation of the 9th testing data can be shown in Table 6, and the count of the 10th testing data can be shown in Table 7.

TABLE 5. Calculation Results of Euclidean Distance for the 8th Data

No	Distance		Class
1	d(x8, x1)	104.1926772	Benign
2	d(x8, x2)	49422.04809	Mirai_ack
3	d(x8, x3)	36635.28506	Mirai_ack
4	d(x8, x4)	47616.04649	Mirai_ack
5	d(x8, x5)	68.24917455	Mirai_scan
6	d(x8, x6)	31.53023299	Mirai_scan
7	d(x8, x7)	95.28565685	Mirai_scan

TABLE 6. Calculation Results of Euclidean Distance for the 9th Data

No	Distance		Class
1	d(x9, x1)	28.06720063	Benign
2	d(x9, x2)	49443.47196	Mirai_ack
3	d(x9, x3)	36656.41549	Mirai_ack
4	d(x9, x4)	47637.28388	Mirai_ack
5	d(x9, x5)	57.03069441	Mirai_scan
6	d(x9, x6)	128.4353453	Mirai_scan
7	d(x9, x7)	43.10280146	Mirai_scan

TABLE 7. Calculation Results of Euclidean Distance for the 9th Data

No	Distance		Class
1	d(x10, x1)	216.7492932	Benign
2	d(x10, x2)	49394.84223	Mirai_ack
3	d(x10, x3)	36608.27336	Mirai_ack
4	d(x10, x4)	47588.92166	Mirai_ack
5	d(x10, x5)	183.0696866	Mirai_scan
6	d(x10, x6)	105.6484925	Mirai_scan
7	d(x10, x7)	210.5725472	Mirai_scan

The next stage is voting by taking the closest training data to the test data as much as K data, then voting for the most dominant class from the class on the training data taken. The voting results will be used to determine the class on the test data.

The results of the Euclidean Distance are sorted from the smallest to the largest. The results of sorting the closest data as much as K for the 8th data, 9th data and 10th data from the Euclidean Distance calculation are shown in Table 8, Table 9 and Table 10.

TABLE 8. Results of the Closest Data Collection of K for the 8th Data

Distance		Total	Class	Rank
6	D(8,6)	31.53023299	Mirai_scan	1
5	D(8,5)	68.24917455	Mirai_scan	2
7	D(8,7)	95.28565685	Mirai_scan	3
1	D(8,1)	104.1926772	Benign	4
3	D(8,3)	36635.28506	Mirai_ack	5

TABLE 9. Results of the Closest Data Collection of K for the 8th Data

Distance		Total	Class	Rank
1	D(9,1)	28.06720063	Benign	1
7	D(9,7)	43.10280146	Mirai_scan	2
5	D(9,5)	57.03069441	Mirai_scan	3
6	D(9,6)	128.4353453	Mirai_scan	4
3	D(9,3)	36656.41549	Mirai_ack	5

TABLE 10. Results of the Closest Data Collection of K for the 8th Data

Distance		Total	Class	Rank
6	D(10,6)	105.6484925	Mirai_scan	1
5	D(10,5)	183.0696866	Mirai_scan	2
7	D(10,7)	210.5725472	Mirai_scan	3
1	D(10,1)	216.7492932	Benign	4
3	D(10,3)	36608.27336	Mirai_ack	5

The next stage of voting to determine the class on the test data based on the most dominant class of the data taken is shown in Table 11.

TABLE 11. K Data Collection Results

Data -i	Class Set	Class Voting
8	{Mirai_scan, Mirai_scan, Mirai_scan, Benign, Mirai_ack}	Mirai_scan
9	{Benign, Mirai_scan, Mirai_scan, Mirai_scan, Mirai_ack}	Mirai_scan
10	{Mirai_scan, Mirai_scan, Mirai_scan, Benign, Mirai_ack}	Mirai_scan

4. Confusion Matrix

In testing the accuracy level, the higher the accuracy, the more relevant the class classification results with the actual class. Before making the Confusion Matrix table, a comparison table of the results of the KNN classification is needed with the actual class. The following table compares the results of the KNN classification with the actual class shown in Table 12.

TABLE 12. Comparison of Prediction Class with Actual Class

No	Class Voting	Actual Class
8	Mirai_scan	Mirai_scan
9	Mirai_scan	Mirai_scan
10	Mirai_scan	Mirai_scan

B. KNN Testing Using Python

At the testing stage, several test data will be classified using the classification model that has been formed. The distance measurement used in the classification process is the Euclidean Distance method, and the number of K is 5. The test data used in the testing phase is separate from the data used during the training.

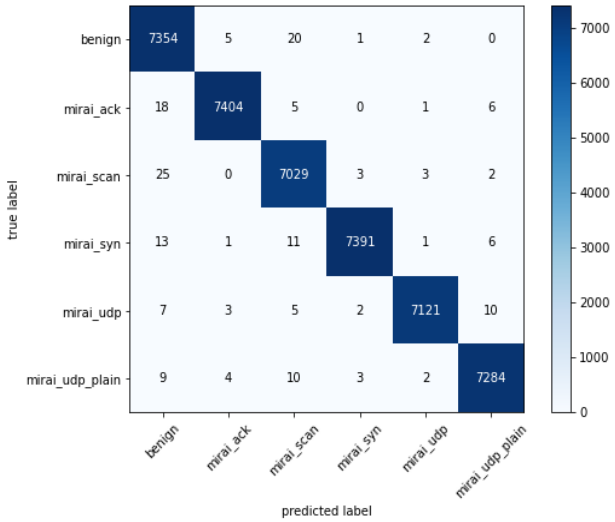

Fig. 2. Confusion Matrix Testing Classification Model

Cecep Suprayogi and M Akbar Marwan, "Classification of Network Traffic Data Mirai Malware Attacks on Internet of Things Devices Using the K-Nearest Neighbor Method," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 4, pp. 39-43, 2022.

The test data used was 36449 data consisting of Benign data of 7382, Mirai ACK data of 7434, Mirai Scan data of 7062, Mirai SYN data of 7423, Mirai UDP data of 7148 and Mirai UDP Plain data of 7312. Measuring the performance of the resulting classification model is carried out with the Confusion Matrix. Figure 2 shows the results of the Confusion Matrix as a result of testing the classification model.

From the Confusion Matrix, as shown in Figure 2, it is known that in the Benign class with a total sample of 7382 correctly predicted data, as many as 7354 data, 5 predictable data as Mirai ACK class, 20 predictable data as Mirai Scan class, 1 predictable data as Mirai SYN class, and 2 predictable data as Mirai UDP class. In the Mirai ACK class, with a total sample of 7434 correctly predicted as many as 7404 data, 18 predicted data as Begin class, 5 predicted data as Mirai Scan class, 1 predictable data as Mirai UDP class, and 6 predictable data as Mirai UDP Plan class. In the Mirai Scan class, with a total sample of 7062 correctly predicted as many as 7029 data, 25 predictable data as Benign class, 3 predicted data as Mirai SYN class, 3 predictable data as Mirai UDP class, and 2 predictable data as Mirai UDP Plan class. In the Mirai SYN class, with a total sample of 7391 correctly predicted as many as 7391 data, 13 predicted data as Benign class, 1 predicted data as Mirai ACK class, 11 predictable data as Mirai Scan class, 1 predictable data as Mirai UDP class, and 6 predictable data as Mirai UDP Plain class. In the Mirai UDP class, with a total sample of 7148 correctly predicted as many as 7121 data, 7 predictable data as Benign class, 3 predictable data as Mirai ACK class, 5 predictable data as Mirai Scan class, 2 predictable data as Mirai SYN class, and 10 predictable data as Mirai UDP Plain class. While in the Mirai UDP Plain class, with a total sample of 7312 correctly predicted as many as 7284 data, 9 predictable data as Benign class, 4 predicted data as Mirai ACK class, 10 predicted data as Mirai SCAN class, 3 predictable data as Mirai SYN class, and 2 predictable data as Mirai UDP class. From the Confusion Matrix, the accuracy of the classification model can be calculated based on Equation (2).

$$\text{Accuracy} = \frac{\text{Calculated Correct Number of Class Data}}{\text{Overall Class Data Count}} \text{ x } 100\%$$

$$Accuracy = \frac{\begin{matrix}(7354+7404+7029\\+7391+7121+7284)\end{matrix}}{\begin{matrix}(7354+5+20+1+2+0+\\18+7404+5+0+1+6+\\25+0+7029+3+3+2+\\13+1+11+7391+1+6+\\7+3+5+2+7121+10+\\9+4+10+3+2+7284)\end{matrix}} x\ 100\%$$

$$Accuracy = \frac{43583}{43761} x100\%$$

$$Accuracy = 99{,}59\%$$

From the test results of the KNN method using the Python programming language by determining the number of K neighbours as many as 5, the number of data predicted to be correct as much as 4358 data divided by the total number of class data as much as 43761 data, then the calculation of the percentage of accuracy obtained in this study got an accuracy rate of 99.59%.

V. CONCLUSION

Based on the results of research in implementing the classification of malware attack traffic data on IoT devices with the KNN method carried out by researcher, then several conclusions can be drawn, namely:

- This study successfully applied the KNN algorithm classification to detect Mirai Malware attacks in IoT device attacks.
- This study used 145868 record data, 102107 data for training data, and 43761 for testing data. From the testing data, 43581 data is predicted to be correct and 178 data are predicted to be wrong by calculating the number of K as much as 5.
- This study successfully tested the confusion matrix classification model using the KNN method to detect malware attacks in IoT device attacks. The results of the tests carried out have a relatively high level of accuracy of 99.59%.

REFERENCES

[1] Gopal, T. S. et al, "Mitigating Mirai Malware Spreading in IoT Environment," *International Conference on Advances in Computing, Communications and Informatics*, pp. 2226–2230, 2018.

[2] Banerjee, M., & Samantaray, S. D, "Network traffic analysis based IoT botnet detection using honeynet data applying classification techniques," *International Journal of Computer Science and Information Security*, 2019.

[3] Churcher, A. et al, "An experimental analysis of attack classification using machine learning in IoT networks," *Sensors (Switzerland)*, 21(2), pp. 1–32, 2021.

[4] Danukusumo, K. P, "Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU," S1 thesis, UAJY, 2017.

[5] Afzaal, M., Usman, M. and Fong, "Tourism mobile app with aspect-based sentiment classification framework for tourist reviews," *IEEE Transactions on Consumer Electronics*, vol. 65, issue 2, pp. 233–242, 2019.

[6] T. Lotlikar, S. Madhavan, S. Andrews, C. M. and J. M, "DoShield Through SDN for IoT Enabled Attacks," *International Conference on Electronics, Communication and Aerospace Technology*, pp. 1499–1504, 2018

[7] Youllia, I. N., Hermana , A. N., & Kharisma, M, "Penerapan Algoritma K-Nearest Neighbor pada Game Pesawat untuk Pembelajaran Matematika Dasar," *MIND Journal*, vol. 4, issue 2, pp. 132-143, 2019.

[8] Sokolova, M., & Lapalme, G, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, issue 4, pp. 427–437. 2009.

[9] Prasetyawan, Danu, Rahmadhan Gatra, "Algoritma K-Nearest Neighbor untuk Memprediksi Prestasi Mahasiswa Berdasarkan Latar Belakang Pendidikan dan Ekonom, "*Jurnal Informatika Sunan Kalijaga*, vol 7, issue 1, pp. 55-67, 2022.

[10] Farokhah, "Implementasi K-Nearest Neighbor untuk Klasifikasi Bunga dengan Ekstraksi Fitur Warna RGB," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol 7, issue 6, pp. 1129-1136, 2020.

[11] Pamela, Sugih, "Pengolahan Data Traffic Pada Perangkat Internet Of Things Dengan Menggunakan Algoritma Random Forest," Sarjana thesis, Universitas Siliwangi, 2019.

[12] Jaramillo, L. E. S, "Malware Detection and Mitigation Techniques: Lessons Learned from Mirai DDOS Attack," *Journal of Information Systems Engineering & Management*, vol. 3, issue 3, 2018.

[13] Gurulakshmi, K. and Nesarani, A, "Analysis of IoT Bots Against DDOS Attack using Machine Learning algorithm," *International Conference on Trends in Electronics and Informatics*, IEEE, (Icoei), pp. 1052-1057, 2018.