Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 11, No. 1, March 2025, pp. 27-41
ISSN: 2338-3070, DOI: 10.26555/jiteki.v11i1.30477

Malware Classification and Detection using Variations of Machine Learning Algorithm Models

Andi Maslan[1], Abdul Hamid[2]
[1]Department of Informatic Engineering, Faculty of Computer Engineering Universitas Putera Batam, Batam, 29432, Indonesia
[2]FG IM Tech, Faculty of Technical and Vocational Education Universiti Tun Hussein Onn Malaysia 86400, Malaysia

ARTICLE INFO

Article history:

Received December 19, 2024
Revised January 10, 2025
Accepted March 05, 2025

Keywords:

Malware;
Machine Learning;
SVM;
KNN;
Neural Network;
Networking

ABSTRACT

Malware attacks are attacks carried out by an attacker by sending malicious codes to various files or even many packages and servers. Therefore, reliable network operations are a factor that needs to be considered to prevent attacks as early as possible in order to avoid more severe system damage. Types of attacks can be Ping of Death, flooding, remote-controlled attacks, UDP flooding, and Smurf Attacks. Attack data was obtained from the ClaMP dataset, which has an unbalanced data set, and has very high noise, so it is necessary to analyze data packets in network logs and optimize feature extraction which is then analyzed statistically with machine learning algorithms. The purpose of the study is to detect, classify malware attacks using a variety of ML Algorithm models such as SVM, KNN and Neural Network and testing detection performance. The research stage starts from pre-Processing, extraction, feature selection and classification processes and performance testing. Training and testing data in the study used a mixed model, namely data division, split model and cross validation. The results of the study concluded that the best algorithm for detecting malware packages is the Neural Network for the Feature Combination category with an accuracy rate of 96.91%, Recall of 97.35% and Precision of 96.78%. So that the study can have implications for cyber experts to be able to prevent malware attacks early. While further research requires a special algorithm to improve malware attack detection, in addition to KNN, SVM and Neural Network. And another research challenge is to focus on feature extraction techniques on datasets that have unbalanced or varied features with the Natural Language Processing (NLP) approach. So this research can be used as a reference for researchers who are conducting research in the same field.

Corresponding Author:

Andi Maslan, Universitas Putera Batam, Indonesia. 29432
Email: andimaslan@puterabatam.ac.id

1. INTRODUCTION

Malware attacks have become a very big risk nowadays because the use of technology is increasing and developing day by day. There are various types of Malware or malicious programs found on the internet. Research shows that Malware has grown exponentially over the last decade, causing huge financial losses to various organizations. Malware is a malicious program or software that proves to be very harmful to the user's device. The user's system can be affected in several ways. The proposed solution uses various machine learning techniques to detect whether or not the file downloaded from the internet contains Malware [1]. The types of Malware available today are Adware, Trojan, Backdoors, Unknown, Multidrop, Rbot, Spam, and Ransomware [2], [3]. This Malware can damage, steal and even delete various types of files on the computer system, thus harming various parties or organizations. In addition, Malware is very dangerous for computer data security

systems and reduces network performance by attacking into computer systems and networks through open ports that are not used in the network system [3]. For example, Trojans and spyware will infect computers in many ways, such as through email, disguised as good software, in the form of links or other files [4], [5]. This Malware can see important data and files and even activities on the user's device on the Android platform [6], [7] and infiltrate through the app store application distribution service and Google Play Store or third parties by disguising themselves as legitimate applications such as video players, games and system utilities [8], [9], [4]. And recently also found a file that is considered Malware that disguises PDF files into PDF, if users are not observant in both types of files then it is likely that this type of Malware will steal mobile banking data, and as a result harm bank customers.

This study aims to use machine learning algorithms with many models so that they can detect Malware accurately and distinguish between malicious and harmless files. In detecting Malware attacks, many algorithms can be used such as decision trees, random forests, support vector machines and this algorithm is very good for classification on cyber-threat actor datasets (CTA) [10]. In addition, research for malware detection can also be done by studying various features of downloaded files such as MD5 hash, Optional Header size, and payload configuration size often called payload. Based on the analysis done on these features, the file will be classified as malicious or harmless. Models are trained on various features that enable them to learn how to classify files. After proper training, the models will be compared with each other based on various criteria. This comparison is done with the help of validation and testing datasets. Finally, the model with the best accuracy will be selected. This process helps to identify all types of Malware that can negatively impact the user's system after being infected. The approach used here will be able to detect Malware such as Adware, Trojans, Backdoors, Unknown, Multidrop, Rbot, Spam, and Ransomware [11], [12].

In addition, machine learning techniques are part of the methods used in detecting traffic anomalies [9], [13]. Machine learning methods are divided into two types: supervised and unsupervised. In supervised learning, training data is labeled according to its class. Data labeling involves marking, for example in a training data set into legitimate and attack classes [14]. Supervised learning techniques learn from labeled input examples and produce a classifier that can be used to map unseen data into one of two pre-defined classes. This classifier can be represented in the form of a set of rules. An intrusion detection system using supervised learning techniques can be equipped to load these rules, filter instances based on different feature values, and classify new instances, for example, into legitimate or attacks [15], [16], [17].

Unsupervised learning techniques find statistical relationships among data instances and classify instances based on their correlations [14], [18], [19]. The technique does not require labeled training data for learning; instead, it will learn from a probabilistic data model [20]. Intrusion detection systems that use unsupervised learning methods are equipped with several statistical parameters, such as learning rate, and which are used to calculate the error measurement between new instances and other data. Instances that are statistically different from the others are considered to belong to another class or anomalies.

Semi-supervised learning can be said to be a combination of the two types of machine learning that have been explained previously [21]. Semi-supervised learning is a type of machine learning that can work using labeled data for small-scale data and unlabeled data for large-scale data. This type of machine learning can be combined with other machine learning methods such as classification, regression, and [22], [23], Semi-supervised learning can be divided into two, namely inductive methods and transductive methods. The inductive method aims to label new data without training data [24], [25]. Apart from the type of machine learning, classification algorithms play an important role in detecting attacks.

Classification algorithms are used to automate and extend the ideas of heuristic-based methods [26], [27], [28]. Classification is the process of declaring an object into one of the previously defined categories [29], [30], [31]. The learning process is a target function that maps each set of attributes x (input) to one of the predetermined classes y. The input is a set of records (training set) of each record consisting of attributes, and one of the attributes is class [32].

From the background and problems that have been described, malware attacks from day to day still exist, because of the large amount of data flowing on the internet, so that data is an important asset that is targeted by cybercrime. In addition, the machine learning approach is still part of AI which is the best technique in detecting malware attacks. The SVM, KNN and Neural Network algorithms are the choice because they are able to identify and process large data, including finding patterns and trends in big data in a relatively short time.

2. METHODS

The method used to detect malware in this study uses various algorithm models by extracting features, then categorizing them into several feature parts with the aim of obtaining a classification of the type of malware attack [33]. The research steps are explained in the following sub-chapters.

2.1. Research Design

It is necessary to arrange the stages of the research to obtain structured and logical results. At the research stage, the methods used to achieve the output produced in this study can be seen in Fig. 1. The following figure illustrates the workflow process from start to finish to provide a more complete understanding of the proposed machine-learning method for malware detection.

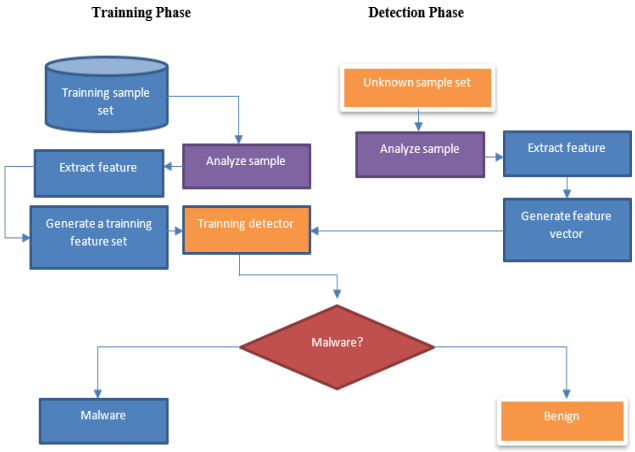

Fig. 1. Research Design

From the research design above, the general steps are explained as follows:

- **Pre-Processing**

Data is stored in the file system as binary code, and the file itself is an unprocessed executable. Research prepares it before research is carried out. Unpacking an executable requires a protected environment, or virtual machine (VM). Special software is used to unpack the compressed executable automatically.

- **Feature Extraction**

Features Datasets of the 20th century often contained tens of thousands of features. Recently, as the number of features has increased, it has become clear that the resulting machine learning models have become overdeveloped. To address this issue, this study builds a smaller set of features from a larger set of features; this technique is used to maintain the same level of accuracy while using fewer features[34]. This study aims to improve existing dynamic and static feature datasets by retaining the most useful features and eliminating unimportant ones for data analysis.

- **Generate feature**

After performing feature extraction, which finds more features, feature selection is performed. Feature selection is used to improve accuracy, simplify the model, and reduce overfitting as it involves selecting features from a set of newly recognized feature qualities. The feature selection technique uses the Weight By Correlation method. which can increase the detection accuracy value because Weight by Correlation weights the attributes by connecting (correlating) one attribute with another attribute. The formula used is as follows:

$$r = \frac{S(x-\bar{x})(y-\bar{y})}{\sqrt{S(x-\bar{x})^2 S(y-\bar{y})^2}} \quad (1)$$

where x is feature, y is feature class, $\bar{x}$ is average feature value of the observed, $\bar{y}$ is average feature value of the expected.

If a coefficient value is -1 (strongly negative) between two variables, it indicates that the variable's linear relationship is perfectly negative, i.e., features are unrelated.

(i) if a coefficient value is 0, it indicates that there is no linear relationship between variables X and Y. However, it does not yet mean that features X and Y are independent.

If a coefficient value is 1 (strongly positive), it indicates that the linear relationship between such variables is perfectly positive; as an example, Pearson's coefficient value for a feature itself (i.e., between variables X and X, or Y and Y) is 1. Each feature has a weight calculated in the order of the largest to the smallest order, ranging from 0.99 to 0.10 [35].

- **Training and Testing**

The training set is a part of the malware dataset that is trained to predict a malware or benign package by running the function of an ML algorithm, in this study using 3 algorithms namely SVM, Neural Network and

KNN. While the Test set is a part of the dataset to see the accuracy of malware detection by considering the True-Positive (TP) value indicating the number of correct predictions of Malware, False-Positive (FP) is the number of malware prediction errors in Normal, False-Negative (FN) is the number of Normal prediction errors in Malware, and At the same time, True-Negative (TN) is the number of correct predictions on Normal.

2.2. Classification approaches in malware detection

This study evaluates using the confusion matrix method. The confusion matrix helps analyze how well the classifier recognizes tuples from different classes [19]. True-Positive and True-Negative values provide information when the classifier classifies the data correctly. Conversely, false-positive and false-negative methods provide information when the classifier is wrong in classifying the data [20]. In this study, normal traffic is indicated by positive values, and negative values indicate malware traffic. The matrix table is shown in Table 1.

Table 1. Confusion Matrix

Actual \ Prediction	DDoS	Normal
Malware	TP	FN
Normal	FP	TN

True-Positive (TP) shows the number of correct predictions of Malware. False-Positive (FP) is the number of malwares mispredictions in Normal. False-Negative (FN) is the number of Normal mispredictions in Malware. At the same time, True-Negative (TN) is the number of correct predictions in Normal.

From the confusion matrix, accuracy, precision, recall, and F-measure can be measured to analyze the performance of machine learning algorithms in classifying and detecting malware attacks with the following equation.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$\text{Precession} = \frac{TP}{TP+FN} \tag{3}$$

$$\text{Recall} = \frac{TN}{TN+FN} \tag{4}$$

$$\text{F-Measure} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \tag{5}$$

Based on the value between False Positive Rate and True Positive Rate, classification performance analysis can be carried out based on Receiver Operating Characteristics (ROC). According to [36], output curves are also used to evaluate classification models, one of which is the ROC curve. The ROC curve shows the trade-off between recall and false alarm rate. According to the ROC curve, a high recall value can also potentially cause a higher false alarm rate.

- **Support Vector Machine (SVM)**

Support Vector Machine is a new approach to supervised pattern classification that has been successfully applied to a variety of pattern recognition problems [11], [37]. Support vector machines are training algorithms for learning classification and regression rules from data. SVMs are best suited to work efficiently with high-dimensional feature spaces [38]. SVMs are based on strong mathematical foundations, resulting in simple yet powerful algorithms. The standard SVM algorithm builds a binary classifier. A simple way to build a binary classifier is to construct a hyperplane that separates class members from non-members in the input space. SVMs also find nonlinear decision functions in the input space by mapping the data into a higher-dimensional feature space and separating them using a maximum margin hyperplane [39]. The system automatically identifies a subset of informative points called support vectors and uses them to represent a sparse separating hyperplane, a linear combination of these points. Finally, SVM solves a simple convex optimization problem. These parameters are obtained by solving the following optimization problem using Support Vector Machine as follows:

$$y_i(x_i \cdot w + b) \geq 1 - \varepsilon i \tag{6}$$

In the equation above, it can be assumed that the hyperplane completely separates the two classes. However, the two classes are not always perfectly separated. This causes the constraints in the equation not to be met and the optimization not to be met. To overcome this problem, SVM is reformulated using the soft margin technique [36] in the following equation.

$$\min w = \tau(w) = \frac{1}{2\|w\|^2} + c\sum_{j=1} \xi_i \tag{7}$$

where, W_i is vector (W_0, W_1, W_2 , W_3……W_m), b is bias term (W_0), X is variable, εi(εi> 0) is Slack variable.

The parameter C is chosen to control the trade-off between the margin and the classification error or the error value in the classification. The parameter C is determined by trying several values and evaluating their effect on the accuracy achieved by the SVM, for example through Cross Validation (CV) [40],[41]. A very large value of C will give a more significant penalty for classification errors.

In general, the data problem is not linearly separable in the input space, and the soft margin SVM cannot find a separator in the hyperplane so it cannot have high accuracy and cannot be generalized well. Therefore, a kernel is needed to transform the data into a higher-dimensional space called the kernel space, which helps to separate the data linearly. The most commonly used kernel functions are Linear, Polynomial and Radial Basis Function (RBF) kernels [42].

- **Neural Network (NN)**

Neural Network is the most widely used classification of neural networks [43]. Artificial neural networks are general-purpose, flexible, nonlinear models consisting of multiple units arranged in layers. The complexity of a NN can be changed by varying the number of layers and the number of units in each layer. Given sufficient units and hidden data, it has been shown that NNs can approximate almost any function with desired accuracy. In other words, NNs are universal approximators. NNs are valuable tools for problems where one has little or no knowledge of the relationship between the input vectors and the corresponding outputs. Neural network models can achieve deep learning. In other words, neural network models can capture much deeper characteristics of traffic [44].

Although neural networks are considered as one of the machine learning methods, neural network models are different from conventional machine learning models that can only learn shallow features.

In general, NN is divided into two parts: single, layer perceptron classifier and multilayer perceptron classifier; each part has a different input, hidden layer, and output [6]. For example, the NN formula used is as follows:

$$Z = Bias + W_1X_1 + W_2X_2 + \cdots + W_nX_n \tag{8}$$

where, Z is symbols for denotation of graphical representation of JST, W_1 is Beta weight or coefficient, X_1 is independent variable or input, and Bias or intercept (W_0).

Three steps must be performed in any neural network:

(i) Input variables and linear combination equations above for$Z = W_0 + W_1X_1 + W_2X_2 + \cdots + W_nX_n$ Calculate the output or predicted value Y, which is called $Ypred$.

(ii) Calculate the loss or error term. The error term is the deviation of the actual value from the predicted value.

(iii) Minimize the loss function or error term.

Computing the neural Network's output can help converge to the optimal solution of the minimum error term. The output layer node depends on the previous hidden layer, which is derived from the previous hidden layer at that node and derived from the input variables. This middle-hidden layer automatically renders the features created by the Network and does not need to take the features [45] explicitly.

- ***K-Nearest Neighbour (KNN)***

The K-Nearest Neighbor algorithm is a method that uses a supervised algorithm. K-Nearest Neighbor includes instance-based learning groups [46]. The K-nearest neighbor algorithm is simple and works based on the similarity of the testing sample with the training sample to determine the K-nearest neighbor. K-Nearest Neighbor searches for groups of objects in the training data that are closest (similar) to objects in the new data or testing data. K-Nearest Neighbor is a simple classification technique but has good results. In general, to determine the distance between two objects x and y, the Euclidean distance formula is used in the following equation:

$$dxy = \sqrt{\sum_{i=1}^{n} (x_1 - y_1)^2} \tag{9}$$

where the matrix D(a,b) is the scalar distance of the two vectors a and b of the d-dimensional matrix. D is proximity. X is training data, Y is testing data, n is number of individual attributes between 1 to d. N, f is attributes of functional similarity between case and case, I is Individual attributes between 1 to n.

Researchers such as [47], [16], [43] have used this algorithm to predict and classify labeled data.

From the three algorithms that have been described, the computational performance of the algorithm has its own advantages and disadvantages [48]. For SVM, it has fast computational performance, KNN is slow and Neural network is slightly faster than KNN, because neural network uses forward propagation. The level of detection accuracy can be seen from the results of this study.

2.3. Data Collection Techniques

A data collection technique related to malware attack detection is needed to support the research data. The data collection technique is carried out by collecting historical data related to Malware. This dataset consists of a collection of Portable Executable (PE) execution files, including examples of Malware and non-malware files [8]. This dataset was collected from the ClaMP malware dataset [49]. The dataset consists of 5,184 samples. The samples consist of 2,683 malware samples and 2,501 benign samples extracted from windows files. In addition, Clamp Malware has 4 main features: Image DoS Header, File_Header and Optional Header [50].

3. RESULTS AND DISCUSSION

In this section, the research results are explained and presented in the form of images, graphs, tables, etc.

3.1. Dataset

The following are the results of the feature collection for the ClaMP malware dataset shown in Table 2.

Table 2. Feature Details

No	Feature	No	Feature	No	Feature
1	e_cblp	26	SizeOfCode	51	OH_DLLchar6
2	e_cp	27	SizeOfInitializedData	52	OH_DLLchar7
3	e_cparhdr	28	SizeOfUninitializedData	53	OH_DLLchar8
4	e_maxalloc	29	AddressOfEntryPoint	54	OH_DLLchar9
5	e_sp	30	BaseOfCode	55	OH_DLLchar10
6	e_lfanew	31	BaseOfData	56	SizeOfStackReserve
7	NumberOfSections	32	ImageBase	57	SizeOfStackCommit
8	CreationYear	33	SectionAlignment	58	SizeOfHeapReserve
9	FH_char0	34	FileAlignment	58	SizeOfHeapCommit
10	FH_char1	35	MajorOperatingSystemVersion	60	LoaderFlags
11	FH_char2	36	MinorOperatingSystemVersion	61	sus_sections
12	FH_char3	37	MajorImageVersion	62	non_sus_sections
13	FH_char4	38	MinorImageVersion	63	packer
14	FH_char5	39	MajorSubsystemVersion	64	packer_type
15	FH_char6	40	MinorSubsystemVersion	65	E_text
16	FH_char7	41	SizeOfImage	66	E_data
17	FH_char8	42	SizeOfHeaders	67	filesize
18	FH_char9	43	CheckSum	68	E_file
19	FH_char10	44	Subsystem	69	fileinfo
20	FH_char11	45	OH_DLLchar0	70	class
21	FH_char12	46	OH_DLLchar1		
22	FH_char13	47	OH_DLLchar2		
23	FH_char14	48	OH_DLLchar3		
24	MajorLinkerVersion	49	OH_DLLchar4		
25	MinorLinkerVersion	50	OH_DLLchar5		

3.2. Extract Feature

From the total features that have been obtained, the next step is to group the main features, with the aim of making it easier to classify the features as follows. Table 3 explains the dataset feature categories in the main features with the aim of making it easier to select features in detecting DDoS type malware, file type and selected feature types that fall into the ClaMP feature dataset category. The purpose of categorizing features is to avoid overfitting between features in each attribute.

3.3. Data Explores

To facilitate the data processing process, here are examples of data types for each feature (Table 4). From the graph in Fig. 2, it is clear that there is a strong relationship between the features in the malware dataset, as attributes that can determine whether the data contains malware files or not. It can be seen from the bright diagonal line in the middle indicating that the elements on the i, i axes have higher values, indicating the relationship of the variables with themselves. And a detailed analysis that 4 features on the x and y axes have a balanced correlation value or a value of 1 indicating that the feature can determine whether or not a malware is present. Meanwhile, the distribution of Malware based on file size can be seen in the following graph in Fig.

3. From the graph in Fig. 3, it is explained that in general Malware can be detected based on the file size which is abnormal from the others.

Table 3. Category Feature

No	Main features	Dataset features	No of features
1	DDoS Feature	e_cblp, e_maxalloc, e_sp, e_lfanew, SizeOfCode, SizeOfInitializedData, AddressOfEntryPoint, BaseOfCode, BaseOfData, CheckSum, SizeOfStackReserve, SizeOfStackCommit, SizeOfHeapReserve, SizeOfHeapCommit, E_text, E_data, filesize, E_file	18
2	File Feature	FH_char0, FH_char1, FH_char2, FH_char3, FH_char4, FH_char5, FH_char6, FH_char7, FH_char8, FH_char9, FH_char10, FH_char11, FH_char12, FH_char13, FH_char14, OH_DLLchar0, OH_DLLchar1, OH_DLLchar2, OH_DLLchar3, OH_DLLchar4, OH_DLLchar5, OH_DLLchar6, OH_DLLchar7, OH_DLLchar8, OH_DLLchar9, OH_DLLchar10	26
3	Optional Feature	e_cp, e_cparhdr, NumberOfSections, CreationYear, MajorLinkerVersion, MinorLinkerVersion, SizeOfUninitializedData, ImageBase, SectionAlignment, FileAlignment, MajorOperatingSystemVersion MinorOperatingSystemVersion, MajorImageVersion, MinorImageVersion, MajorSubsystemVersion, MinorSubsystemVersion, SizeOfImage, SizeOfHeaders, Subsystem, LoaderFlags, sus_sections, non_sus_sections, packer, packer_type, fileinfo	25

Table 4. Feature Data Type

No	Feature	Type Data
1	e_cblp	float64
2	e_cp	float64
3	e_cparhdr	float64
4	e_maxalloc	float64
5	e_sp	float64
..		...
65	E_data	float64
66	filesize	float64
69	E_file	float64
70	fileinfo	float64
	class	float64

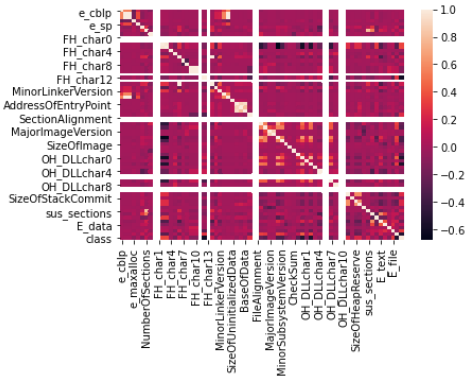

Fig. 2. Visualizes of Data

3.4. Feature Selection

The results of feature selection in this study used the Weight by Correlation method, with the following results in Fig. 4. With details of the correlation value sequence in Table 5. Based on the graph Fig. 4 and Table 5, the best features in detecting malware attacks are FH_char12, OH_DLLchar2, OH_DLLchar0 with respective weight correlation values of 0.6081, 0.5428 and 0.5189, meaning that this feature is very effective in determining whether a package is malicious or not.

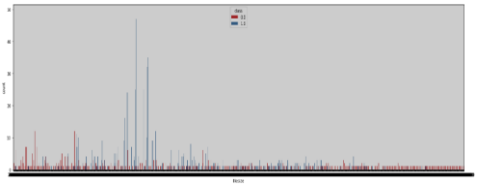

Fig. 3. Data Packet Pattern for File Size Feature

Table 5. Feature Correlation Value

No	Correlations	Feature
1	0.6081	FH_char12
2	0.5428	OH_DLLchar2
3	0.5189	OH_DLLchar0
4	0.5140	fileinfo
5	0.5119	FH_char0
6	0.3526	FH_char3
7	0.3476	FH_char2
8	0.3400	MinorSubsystemVersion
9	0.3321	MajorSubsystemVersion
10	0.3073	E_file
11	0.2838	MinorOperatingSystem Version
12	0.2824	Subsystem
13	0.2423	ImageBase
14	0.2408	SizeOfStackReserve
15	0.2318	SizeOfHeapReserve
16	0.2316	SizeOfHeapCommit
17	0.2310	MajorOperatingSystem Version
18	0.1971	OH_DLLchar7
19	0.1862	sus_sections
20	0.1848	E_text
21	0.1630	FH_char14
22	0.1614	FH_char6
23	0.1448	FH_char5
24	0.1364	SizeOfImage
25	0.1361	e_lfanew
26	0.1240	NumberOfSections
27	0.1183	filesize
28	0.1080	FH_char9
29	0.1080	FH_char10
30	0.0992	MinorLinkerVersion
31	0.0968	E_data
32	0.0922	MajorLinkerVersion
33	0.0897	OH_DLLchar1
34	0.0881	FH_char8
35	0.0872	OH_DLLchar4
36	0.0759	e_maxalloc
37	0.0599	BaseOfData
38	0.0557	non_sus_sections
39	0.0523	OH_DLLchar8
40	0.0480	SizeOfHeaders
41	0.0478	packer
42	0.0432	BaseOfCode
43	0.0370	AddressOfEntry Point
44	0.0310	CreationYear
45	0.0296	LoaderFlags
46	0.0264	SizeOfStackCommit
47	0.0242	SizeOfUninitialized Data
48	0.0227	FH_char7
49	0.0199	SizeOfCode
50	0.0186	e_cp
51	0.0186	e_cparhdr
52	0.0183	e_sp
53	0.0158	ï»¿e_cblp
54	0.0155	SizeOfInitialized Data
55	0.0145	FH_char4
56	0.0145	OH_DLLchar9
57	0.0132	OH_DLLchar3
58	0.0121	CheckSum
58	0.0116	MajorImageVersion
60	0.0056	MinorImageVersion

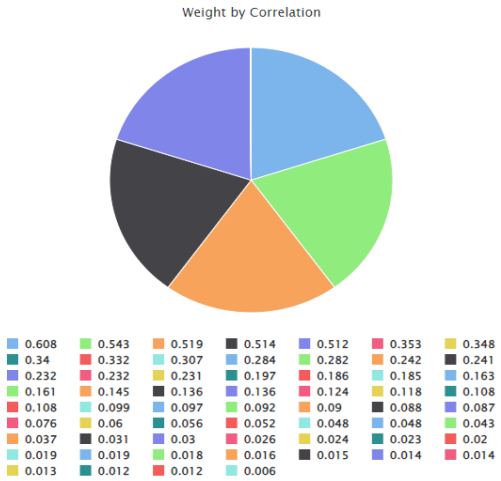

Fig. 4. Weight by Correlation Value

3.5. Trainning dan Testing Data

The results of the training and testing data in this study used a mixed model, with data division using the split and cross-validation models. This is done to help measure the extent to which the model can generalize the data, In addition, the training data is limited or not enough. The results of the training data and the split command 75: 25, from a total of 5,184 data.

3.6. Model Making

- **KNN Algorithm**

Based on the available dataset, a model is needed to produce a good level of accuracy in detecting malware attacks. The models used in the study are KNN, SVM, and Neural Network. From the model that has been worked on, the results of the model are able to see the confusion matrix table with the aim of knowing True Positive, True Negative, False Positive and False Negative as follows in Table 6.

Table 6. DDoS Detection Feature

	true Normal	true Malware	class precision
pred. Normal	2263	235	90.59%
pred. Malware	225	2487	91.70%
class recall	90.96%	91.37%	

From the Table 6, it is explained that the normal data package for Malware Feature with the number of features as many as 18 which were detected as normal packages as many as 2263 packages and malware packages as many as 2487 packages, and normal data packages were detected as malware as many as 225 and malware packages were detected as normal packages as many as 235 with an accuracy level reaching 91.71%. The algorithm used to detect whether data packages are normal or not uses the KNN algorithm.
From the Table 7, it is explained that the normal data package for the File feature with the number of features as many as 26 which were detected as normal packages as many as 2470 packages and malware packages as many as 822 packages, and normal data packages were detected as malware as many as 18 and malware packages were detected as normal packages as many as 1900 with an accuracy level reaching 63.19%. The algorithm used to detect whether data packages are normal or not uses the KNN algorithm.

Table 7. File Feature

	true Normal	true Malware	class precision
pred. Normal	2470	1900	56.52%
pred. Malware	18	822	97.86%
class recall	99.28%	30.20%	

From the Table 8, it is explained that the normal data package for Optional Feature with the number of features as many as 24 which were detected as normal packages as many as 2407 packages and malware packages as many as 2077 packages, and normal data packages were detected as malware as many as 81 and malware packages were detected as normal packages as many as 645 with an accuracy level reaching 86.07%. The algorithm used to detect whether data packages are normal or not uses the KNN algorithm.

Table 8. Optional Feature

	true Normal	true Malware	class precision
pred. Normal	2407	645	78.87%
pred. Malware	81	2077	96.25%
class recall	96.74%	76.30%	

From the Table 9, it is explained that the normal data package for the Feature Combination with a total of 69 features detected as normal packages is 2267 packages and, malware packages are 2485 packages, and normal data packages are detected as malware as many as 221 and malware packages are detected as normal packages as many as 237 with an accuracy level reaching 91.21%. The algorithm used to detect whether data packages are normal or not uses the KNN algorithm.

Table 9. Feature Combination

	true Normal	true Malware	class precision
pred. Normal	2267	237	90.54%
pred. Malware	221	2485	91.83%
class recall	91.12%	91.29%	

From the Table 10, it can be concluded that the combination of features is the best dataset in detecting malware attacks using the KNN algorithm compared to other features categorized as DDoS Features, File Features and Optional features.

Table 10. Malware Feature Combination

Dataset	Category Attribute	No of Feature	Accuracy	Recall	Precision
ClaMP Malware	DDoS Feature	18	91.17	91.37	91.71
	File Feature	26	63.19	30.20	97.93
	Optional Feature	24	86.07	76.31	96.25
ClaMP Malware	Combinasi feature	68	91.21	91.29	91.85

- **SVM Algorithm**

The results of the Machine Learning model performance evaluation in detecting Malware attacks are as follows. Table 11, it is explained that the normal data packets for DDoS Features with the number of features as many as 18 were detected as normal packets as many as 2176 packets and malware packets as many as 2144 packets, and normal data packets were detected as malware as many as 312 and malware packets were detected as normal packets as many as 578 with an accuracy level reaching 82.92%. The algorithm used to detect whether data packets are normal or not uses the SVM algorithm.

Table 11. DDOS Feature

	true Normal	true Malware	class precision
pred. Normal	2176	578	79.01%
pred. Benign	312	2144	87.30%
class recall	87.46%	78.77%	

From the Table 12, it is explained that the normal data package for File Feature with the number of features as many as 26 which were detected as normal packages as many as 1997 packages and malware packages as many as 2575 packages, and normal data packages were detected as malware as many as 491 and malware packages were detected as normal packages as many as 147 with an accuracy level reaching 87.75%. The algorithm used to detect whether data packages are normal or not uses the SVM algorithm.

Table 12. Feature File

	true Normal	True Malware	class precision
pred. Normal	1997	147	93.14%
pred. Malware	491	2575	83.99%
class recall	80.27%	94.60%	

From the Table 13, it is explained that the normal data package for File Feature with the number of features as many as 24 which were detected as normal packages as many as 1949 packages and malware packages as many as 2144 packages, and normal data packages were detected as malware as many as 539 and malware packages were detected as normal packages as many as 578 with an accuracy level reaching 87.75%. The algorithm used to detect whether data packages are normal or not uses the SVM algorithm.

Table 13. Optional Feature

	true Normal	true Malware	class precision
pred. Normal	1949	578	77.13%
pred. Malware	539	2144	79.91%
class recall	78.34%	78.77%	

From the Table 14, it is explained that the normal data package for Feature Combination with a total of 69 features detected as normal packages is 1879 packages and malware packages are 2712 packages, and normal data packages are detected as Malware as much as 609 and Malware packages are detected as normal packages as much as 10 with an accuracy level reaching 88.12%. The algorithm used to detect whether data packages are normal or not uses the SVM algorithm.

Table 14. Feature Combination

	true Normal	true Malware	class precision
pred. Normal	1879	10	99.47%
pred. Malware	609	2712	81.66%
class recall	75.52%	99.63%	

From the Table 15, it can be concluded that the combination of features is the best dataset in detecting malware attacks using the KNN algorithm compared to other features categorized as DDoS Features, File Features and Optional Features.

Table 15. SVM Algorithm Feature Combination

Dataset	Category Attribute	Number of Feature	Accuracy	Recall	Precesion
ClaMP Malware	DDoS Feature	18	82.92	78.76	87.28
	File Feature	26	87.75	85.32	94.60
	Optional Feature	24	78.56	78.77	79.93
ClaMP Malware	Combines feature	68	88.12	99.63	81.70

- **Neural Network Algorithm**

The results of the Machine Learning model performance evaluation in detecting Malware attacks are as follows. Table 16, it is explained that the normal data packet for DDoS Feature with the number of features as many as 18 which were detected as normal packets as many as 2282 packets and malware packets as many as 2509 packets, and normal data packets were detected as malware as many as 206 and malware packets were detected as normal packets as many as 213 with an accuracy level reaching 91.96%. The algorithm used to detect whether data packets are normal or not uses the Neural Network algorithm.

Table 16. DDoS File Algorithm Neural Network

	true Normal	true Malware	class precision
pred. Normal	2282	213	91.46%
pred. Malware	206	2509	92.41%
class recall	91.72%	92.17%	

From the Table 17, it is explained that the normal data package for File Feature with the number of features as many as 26 which are detected as normal packages as many as 2264 packages and malware packages as many as 2600 packages, and normal data packages are detected as malware as many as 224 and malware packages are detected as normal packages as many as 122 with an accuracy level reaching 93.36%. The algorithm used to detect whether data packages are normal or not uses the Neural Network algorithm.

Table 17. Feature File

	true Normal	true Malware	class precision
pred. Normal	2264	122	94.89%
pred. Malware	224	2600	92.07%
class recall	91.00%	95.52%	

From the Table 18, it is explained that the normal data package for Optional Feature with the number of features as many as 24 which were detected as normal packages as many as 2135 packages and malware packages as many as 2419 packages, and normal data packages were detected as malware as many as 353 and malware packages were detected as normal packages as many as 303 with an accuracy level reaching 87.41%. The algorithm used to detect whether data packages are normal or not uses the Neural Network algorithm.

Table 18. Optional Feature

	true Normal	true Malware	class precision
pred. Normal	2135	303	87.57%
pred. Malware	353	2419	87.27%
class recall	85.81%	88.87%	

From the Table 19, it is explained that the normal data package for the Feature Combination with a total of 69 features detected as normal packages is 2399 packages and malware packages are 2650 packages, and normal data packages are detected as Malware as much as 89 and Malware packages are detected as normal packages as much as 72 with an accuracy level reaching 96.91%. The algorithm used to detect whether data packages are normal or not uses the Neural Network algorithm.

Table 19. Feature Combination

	true Normal	true Malware	class precision
pred. Normal	2399	72	97.09%
pred. Malware	89	2650	96.75%
class recall	96.42%	97.35%	

From the Table 20, it can be concluded that the combination of features is the best dataset in detecting malware attacks using the KNN algorithm compared to other features categorized as DDoS Features, File Features and Optional Features.

Table 20. Combination Of Feature Algorithm Neural Network

Dataset	Category Attribute	Number of Feature	Accuracy	Recall	Precision
ClaMP Malware	DDoS Feature	18	91.96	92.17	92.48
	File Feature	26	93.36	95.52	92.11
	Optional Feature	24	87.41	88.87	87.76
ClaMP Malware	Combines feature	68	96.91	97.35	96.78

3.7. Compare Algorithm Result

The results of the Machine Learning model performance evaluation in detecting Malware attacks are as follows. Based on the Table 21, the best algorithm in detecting a malware package or not is the Neural Network for the feature combination category with an accuracy level of 96.91%, Recall of 97.35% and Precision of 96.78%. While the KNN algorithm specifically for the feature file category has a low accuracy rate of 63.19% compared to other algorithms, this is because KNN is not suitable for large datasets and the data distribution must be normal or each data needs to be scaled, so that the results are better. In addition, the neural network algorithm is the best algorithm in this study, because the algorithm is able to perform deep classification [34] based on the hidden layer on each feature.

Table 21. Algorithm Comparison Results

Algorithm	Category Attribute	Number of Features	Accuracy	Recall	Precision	Training and testing execution time with Cross Validation (CV) seconds					
						Cv6	Cv7	Cv8	Cv9	Cv10	avg
KNN	DDoS Feature	18	91.17	91.37	91.71	0.03	0.02	0.02	0.03	0.02	0.024
	File Feature	26	63.19	30.20	97.93	0.03	0.02	0.03	0.03	0.04	0.030
	Optional Feature	24	86.07	76.31	96.25	0.02	0.02	0.03	0.03	0.03	0.026
	All Feature	68	91.21	91.29	91.85	0.09	0.12	0.11	0.1	0.11	0.106
SVM	DDoS Feature	18	82.92	78.76	87.28	0.28	0.49	0.31	0.44	0.46	0.396
	File Feature	26	87.75	85.32	94.60	0.19	0.27	0.28	0.29	0.22	0.25
	Optional Feature	24	78.56	78.77	79.93	1.49	1.26	1.16	1.34	2.24	1.498
	All Feature	68	88.12	99.63	81.70	0.47	0.22	0.4	0.45	0.54	0.416
Neural Network	DDoS Feature	18	91.96	92.17	92.48	1.22	1.15	1.36	1.35	1.53	1.322
	File Feature	26	93.36	95.52	92.11	2.55	1.21	1.23	2.25	2.35	1.918
	Optional Feature	24	87.41	88.87	87.76	0.48	1.3	1.36	1.25	2.03	1.284
	All Feature	68	96.91	97.35	96.78	6.09	9.47	9.35	9.15	10.16	8.844

It can be concluded that each algorithm has its own advantages and disadvantages, the KNN algorithm excels in terms of performance with good processing speed, but the accuracy level is slightly low. For the SVM algorithm, it has stable speed performance with a fairly good level of accuracy, while the performance of the Neural Network algorithm has a fairly long processing speed performance but has a very high level of accuracy.

4. CONCLUSION

The conclusion of the research related to Malware Attack Detection is that the number of data packages in this dataset is 5212 with a total of 69 attributes and two classes. The research stages start from dataset preparation, feature extraction, Data Exploration, feature selection, training and testing and model creation. This model is created using three machine learning algorithms. The results of the feature classification on the dataset can show the performance of each algorithm in detecting Malware. The best algorithm in detecting malware packages is the Neural Network for the feature combination category with an accuracy level of 96.91%, Recall 97.35% and Precision 96.78%.Meanwhile, further malware detection research can be focused on file headers, especially binary file types, using genetic algorithms. The reason the analysis focuses on binary file headers is because the computing process is faster.

Acknowledgments

We would like to thank the University Tun Hussein Onn Malaysia and Universitas Putera Batam for their contributions through internal research and funding for the successful publication of this paper.

REFERENCES

[1] A. Mehrban and P. Ahadian, "Malware Detection in IoT Systems using Machine Learning Techniques," *International Journal of Wireless & Mobile Networks*, vol. 15, no. 6, pp. 13–23, 2023, https://doi.org/10.5121/ijwmn.2023.15602.

[2] A. Kamboj, P. Kumar, A. K. Bairwa, and S. Joshi, "Detection of malware in downloaded files using various machine learning models," *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 81–94, 2023, https://doi.org/10.1016/j.eij.2022.12.002.

[3] A. R. Damanik, H. B. Seta, and T. Theresiawati, "Analisis Trojan Dan Spyware Menggunakan Metode Hybrid Analysis," *Jurnal Ilmiah Matrik*, vol. 25, no. 1, pp. 89–97, 2023, https://doi.org/10.33557/jurnalmatrik.v25i1.2327.

[4] L. M. Kadhum, A. Firdaus, S. I. Hisham, W. Mushtaq, and M. F. A. Razak, "Features, Analysis Techniques, and Detection Methods of Cryptojacking Malware: A Survey," *International Journal on Informatics Visualization*, vol. 8, no. 2, pp. 891–896, 2024, https://doi.org/10.62527/joiv.8.2.2725.

[5] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," *Applied Sciences (Switzerland)*, vol. 9, no. 11, 2019, https://doi.org/10.3390/app9112375.

[6] I. Ben, A. Ouahab, L. Elaachak, and M. Bouhorma, "Image-Based Malware Classification Using Multi-layer Perceptron Image-Based Malware Classification Using Multi-layer Perceptron," *Intelligent Systems and Security: Proceedings of NISS* pp. 453-464, 2021, https://doi.org/10.1007/978-981-16-3637-0.

[7] D. Stiawan, S. M. Daely, A. Heryanto, N. Afifah, M. Y. Idris, and R. Budiarto, "Ransomware detection based on opcode behaviour using k-nearest neighbours algorithm," *Information Technology and Control*, vol. 50, no. 3, pp. 495–506, 2021, https://doi.org/10.5755/j01.itc.50.3.25816.

[8] J. Jiang and F. Zhang, "Detecting Portable Executable Malware by Binary Code Using an Artificial Evolutionary Fuzzy LSTM Immune System," *Security and Communication Networks*, p. 3578695 2021, https://doi.org/10.1155/2021/3578695.

[9] P. Udayakumar, S. Yalamati, L. Mohan, M. J. Haque, G. Narkhede, and K. M. Bhashyam, "Android malware detection using GIST based machine learning and deep learning techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 2, pp. 1244–1252, 2024, https://doi.org/10.11591/ijeecs.v35.i2.pp1244-1252.

[10] E. Irshad and A. Basit Siddiqui, "Cyber threat attribution using unstructured reports in cyber threat intelligence," *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 43–59, 2023, https://doi.org/10.1016/j.eij.2022.11.001.

[11] J. A. Mata-Torres, E. Tello-Leal, J. D. Hernandez-Resendiz, and U. M. Ramirez-Alcocer, "Evaluation of Machine Learning Techniques for Malware Detection," *Intelligent Systems Reference Library*, vol. 226, pp. 121–140, 2023, https://doi.org/10.1007/978-3-031-08246-7_6.

[12] A. Martín, R. Lara-Cabrera, and D. Camacho, "Android malware detection through hybrid features fusion and ensemble classifiers: The AndroPyTool framework and the OmniDroid dataset," *Information Fusion*, vol. 52, pp. 128–142, 2019, https://doi.org/10.1016/j.inffus.2018.12.006.

[13] C. Singha, V. K. Rana, Q. B. Pham, D. C. Nguyen, and E. Łupikasza, "Integrating machine learning and geospatial data analysis for comprehensive flood hazard assessment," *Environmental Science and Pollution Research*, vol. 31, no. 35, pp. 48497–48522, 2024, https://doi.org/10.1007/s11356-024-34286-7.

[14] K. K. Verma, B. M. Singh, and A. Dixit, "A review of supervised and unsupervised machine learning techniques for suspicious behavior recognition in intelligent surveillance system," *International Journal of Information Technology (Singapore)*, vol. 14, no. 1, pp. 397–410, 2022, https://doi.org/10.1007/s41870-019-00364-0.

[15] J. Pavithra and S. Samy, "A Comparative Study on Detection of Malware and Benign on the Internet Using Machine Learning Classifiers," *Mathematical Problems in Engineering*, p. 4893390, 2022, https://doi.org/10.1155/2022/4893390.
[16] G. Farahani, "Feature Selection Based on Cross-Correlation for the Intrusion Detection System," *Security and Communication Networks*, vol. 2020, pp. 1–17, 2020, https://doi.org/10.1155/2020/8875404.
[17] M. Jedh, L. Ben Othmane, N. Ahmed, and B. Bhargava, "Detection of Message Injection Attacks onto the CAN Bus Using Similarities of Successive Messages-Sequence Graphs," *IEEE Transactions on Information Forensics and Security*, vol. 16, no. 1, pp. 4133–4146, 2021, https://doi.org/10.1109/TIFS.2021.3098162.
[18] Y. Bouchlaghem, Y. Akhiat, and S. Amjad, "Feature Selection: A Review and Comparative Study," *E3S Web of Conferences*, vol. 351, no. May, 2022, https://doi.org/10.1051/e3sconf/202235101046.
[19] A. Fricticarani and H. Maksum, "Improving Student Activity and Learning Outcomes by Applying the Jigsaw Type Learning Model in PPHP Skills Study," *Journal of Education Research and Evaluation*, vol. 4, no. 4, p. 296, 2020, https://doi.org/10.23887/jere.v4i4.30240.
[20] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 283–294, 2020, https://doi.org/10.1007/s12065-019-00310-w.
[21] K. Swapna, "Semi-Supervised Machine Learning For Ddos Attack Classification Using Clustering Based," *IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON),* vol. 12, no. 12, pp. 472–478, 2021, https://doi.org/10.1109/UEMCON47517.2019.8993021.
[22] K. Bouzoubaa, Y. Taher, and B. Nsiri, "Predicting DOS-DDOS Attacks: Review and Evaluation Study of Feature Selection Methods based on Wrapper Process," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 132–145, 2021, https://doi.org/10.14569/IJACSA.2021.0120517.
[23] M. Aamir and S. M. Ali Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 4, pp. 436–446, 2021, https://doi.org/10.1016/j.jksuci.2019.02.003.
[24] Suyanto, *Machine Learning Tingkat Dasar dan Lanjut*. Informatika, 2018.
[25] M. Hemmat Esfe, D. Toghraie, F. Amoozadkhalili, and S. Alidoust, "Optimization of accuracy in estimating the dynamic viscosity of MWCNT-CuO/oil 10W40 nano-lubricants," *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 117–128, 2023, https://doi.org/10.1016/j.eij.2022.12.006.
[26] S. Brindha, M. P. Abirami, V. Arjun, B. Logesh, and M. S. P, "Heuristic Approach to Intrusion Detection System," *Int Res J Eng Technol,* vol. 7, no. 3, pp. 377–379, 2020, https://mail.irjet.net/archives/V7/i3/IRJET-V7I381.pdf'.
[27] S. U. Nisa, A. Mahmood, F. S. Ujager, and M. Malik, "HIV/AIDS predictive model using random forest based on socio-demographical, biological and behavioral data," *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 107–115, 2023, https://doi.org/10.1016/j.eij.2022.12.005.
[28] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019, https://doi.org/10.1186/s42400-019-0038-7.
[29] I. Anggraeni and D. M. Akhmad, "Detection and Classification of DDos Attack on Software Defined Network," *Komputasi: Jurnal Ilmiah Ilmu Komputer dan Matematika*, vol. 19, no. 2, pp. 77–86, 2022, https://doi.org/10.33751/komputasi.v19i2.4769.
[30] R. H. Abbas and F. A. E. A. Kareem, "Text Language Identification Using Letters (Frequency, Self-information, and Entropy) Analysis for English, French, and German Languages," *Journal of Southwest Jiaotong University*, vol. 54, no. 4, 2019, https://doi.org/10.35741/issn.0258-2724.54.4.21.
[31] M. B. Al Amin, R. S. Ilmiaty, and A. Marlina, "Flood Hazard Mapping in Residential Area Using Hydrodynamic Model HEC-RAS 5.0," *Geoplanning*, vol. 7, no. 1, pp. 25–36, 2020, https://doi.org/10.14710/geoplanning.7.1.25-36.
[32] J. P. Tanjung, F. C. Tampubolon, A. W. Panggabean, and M. A. A. Nandrawan, "Customer Classification Using Naive Bayes Classifier With Genetic Algorithm Feature Selection," *Sinkron*, vol. 8, no. 1, pp. 584–589, 2023, https://doi.org/10.33395/sinkron.v8i1.12182.
[33] N. Pachhala, S. Jothilakshmi, and B. P. Battula, "Cross-Platform Malware Classification: Fusion of CNN and GRU Models," *International Journal of Safety and Security Engineering*, vol. 14, no. 2, pp. 477–486, 2024, https://doi.org/10.18280/ijsse.140215.
[34] S. Singh, D. Krishnan, V. Vazirani, V. Ravi, and S. A. Alsuhibany, "Deep hybrid approach with sequential feature extraction and classification for robust malware detection," *Egyptian Informatics Journal*, vol. 27, p. 100539, 2024, https://doi.org/10.1016/j.eij.2024.100539.
[35] D. Stiawan *et al.,* "CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020, https://doi.org/10.1109/ACCESS.2020.3009843.
[36] B. Santosa, A. Umam. *Data Mining and Big Data Analytics: Teori dan Implementasi Menggunakan Python & Apache Spark*. Yogyakarta: Penebar Media Pustaka, 2018, https://books.google.co.id/books/about/Data_Mining_dan_Big_Data_Analytics_Teori.html?id=w1nUDwAAQBAJ&redir_esc=y.
[37] J. S. Pimentel, R. Ospina, and A. Ara, "A novel fusion Support Vector Machine integrating weak and sphere models for classification challenges with massive data," *Decision Analytics Journal*, vol. 11, p. 100457, 2024, https://doi.org/10.1016/j.dajour.2024.100457.
[38] C. Gambella, B. Ghaddar, and J. Naoum-Sawaya, "Optimization problems for machine learning: A survey," *European Journal of Operational Research*, vol. 290, no. 3, pp. 807–828, 2021,

https://doi.org/10.1016/j.ejor.2020.08.045.
[39] R. Guido, S. Ferrisi, D. Lofaro, and D. Conforti, "An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review," *Information (Switzerland)*, vol. 15, no. 4, 2024, https://doi.org/10.3390/info15040235.
[40] F. Talpur, I. A. Korejo, A. A. Chandio, A. Ghulam, and M. S. H. Talpur, "ML-Based Detection of DDoS Attacks Using Evolutionary Algorithms Optimization," *Sensors*, vol. 24, no. 5, 2024, https://doi.org/10.3390/s24051672.
[41] F. S. Prity *et al.*, "Machine learning-based cyber threat detection: an approach to malware detection and security with explainable AI insights," *Human-Intelligent Systems Integration*, pp. 1-30, 2024, https://doi.org/10.1007/s42454-024-00055-7.
[42] M. Najafimehr, S. Zarifzadeh, and S. Mostafavi, "A hybrid machine learning approach for detecting unprecedented DDoS attacks," *The Journal of Supercomputing,* vol. 78, no. 6, pp. 8106-8136, 2022, https://doi.org/10.1007/s11227-021-04253-x.
[43] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," *Sustainability (Switzerland)*, vol. 12, no. 3, 2020, https://doi.org/10.3390/su12031035.
[44] A. G. Ismaeel *et al.*, "Traffic Pattern Classification in Smart Cities Using Deep Recurrent Neural Network," *Sustainability (Switzerland)*, vol. 15, no. 19, pp. 1–17, 2023, https://doi.org/10.3390/su151914522.
[45] Z. Chiba, N. Abghour, K. Moussaid, A. El, and M. Rida, "Intelligent and Improved Self-Adaptive Anomaly based Intrusion Detection System for Networks," *International Journal of Communication Networks and Information Security*, vol. 11, no. 2, pp. 312–330, 2019, https://www.proquest.com/openview/7971f25b84138931244408e5e35bd695a/1?pq-origsite=gscholar&cbl=52057.
[46] S. Sadhwani, B. Manibalan, R. Muthalagu, and P. Pawar, "A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques," *Applied Sciences (Switzerland)*, vol. 13, no. 17, 2023, https://doi.org/10.3390/app13179937.
[47] T. Hairani, "Botnet Detection Using K-Nearest Neighbor Algorithm,"*Review Point,* 2018, https://reviewpoint.org/blog/botnet-detection-using-the-k.
[48] L. Hammood, İ. A. Doğru, and K. Kılıç, "Machine Learning-Based Adaptive Genetic Algorithm for Android Malware Detection in Auto-Driving Vehicles," *Applied Sciences (Switzerland)*, vol. 13, no. 9, 2023, https://doi.org/10.3390/app13095403.
[49] K. Kumari and M. Mrunalini, "Detecting Denial of Service attacks using machine learning algorithms," *Journal of Big Data*, vol. 9, no. 1, 2022, https://doi.org/10.1186/s40537-022-00616-0.
[50] H. Al-Khshali and M. Ilyas, "Impact of Portable Executable Header Features on Malware Detection Accuracy," *Computers, Materials and Continua*, vol. 74, pp. 153–178, Aug. 2022, https://doi.org/10.32604/cmc.2023.032182.

BIOGRAPHY OF AUTHORS

Andi Maslan received the degree in Informatics Engineering was taken at the Budi Utomo Institute of Technology Jakarta (2004) and the master's degree in Computer Science (Information System) completed at the STMIK Putera Batam (2011). Completed at doctoral education at Tun Onn Hussein University Malaysia (2024). The author is a lecturer at the University of Putera Batam and has a functional position as assistant professor. His current research interests include networking, Network Security, and artificial intelligence. He can be contacted at email: Lanmasco@gmail.com.

Abdul Hamid received a PhD in Engineering Technology from the Universiti Tun Hussein Onn Malaysia (UTHM) in 2019. He is currently a senior lecturer with the Department of Technology Studies, UTHM Johor Malaysia. He has published 40 academic papers as a first author or a co-author in conference proceedings and international journals. His research interests in applied sciences, engineering and technologies include Smart manufacturing, transportation and society, mechanical engineering, informatics visualisation, and geoinformatics. He can be contacted at email: abdulhamid@uthm.edu.my.