Detailed Analyses and Efficient Identification of Malware Evidence in CLaMP Dataset based on Machine Learning Approaches

[1]M. O. Ayinla
Department of Computer Science
Kwara State College of Education,
Ilorin, Nigeria
Email: mo.ayinla [AT] kwcoeilorin.edu.ng

[2]A. M. Oyelakin
Department of Computer Science
Crescent University,
Abeokuta, Nigeria
Email: moruff.oyelakin [AT] cuab.edu.ng

[3]U. A. Adeniyi
Department of Cyber Security
Airforce Institute of Technology,
Kaduna, Nigeria
Email: adedayousman [AT] afit.edu.ng

[4]K. O. Tajudeen
Department of Computer Science
Al-Hikmah University,
Ilorin, Nigeria
Email: kotajudeen [AT] alhikmah.edu.ng

[5]O. J. Olaleye
Department of Computer Science & Information Technology
Bells University of Technology,
Ota, Nigeria
Email: ojolaleye [AT] bellsuniversity.edu.ng

Abstract—**Malware is a malicious software that is used to launch attacks of different types in computer networks and cyber space. Several signature and machine learning-based approaches have been used for the identification of malware types in the past. However,signature-based detection approaches have been reported to have serious limitations which gave room for machine learning-based malware identification techniques to be more popular. Despite the promises of the ML methods in the identification of malware evidence, some of the ML approaches in literature have poor detection rates which can be as a result of the size and nature of the patterns in the datasets used. This study used a dataset named CLaMP for the training and testing of the malware classification models. Firstly, comprehensive exploratory analyses of the dataset were carried out with a view to understanding the data distributions in it better and make informative decisions on how to pre-process and apply it for malware identification. During the experimentations, two scenarios were established before feeding the data into the learning algorithms. Scenario 1 involves building malware identification model without data cleaning and feature selection while scenario 2 involves the cleaning of the data and selection of promising features for building the models.In scenario 2, Recursive Feature Elimination (RFE) technique was used for selecting the promising attributes which were used to build the two malware classification models. Naive Bayes (NB) and Logistic Regression (LR) algorithms were used for building the models. The hyper parameters of the two selected algorithms were varied and the models tested and validated severally before optimal performances were arrived at. The results of the models were compared based on the selected metrics, namely: accuracy, precision, recall, f1-score and Area Under the Curve (AUC). Experimental results showed that in the scenario 1, where the dataset was not pre-processed and all the attributes were used for the model building, poor results were obtained by both models in all metrics except in recall. However, NB-based malware identification model slightly performed better than LR in all the metrics. It was also discovered that both NB and LR-based malware identification models performed well in scenario 2 when the dataset was pre-processed and promising features were selected using RFE. This study concluded that the detailed exploratory analyses, data cleaning and feature subset selection methods helped in achieving promising results from the malware identification models**

Keywords- Malware Identification; Machine Learning Algorithms; Feature Selection; Windows PE headers

I. INTRODUCTION

Malware of different types are used to launch various attacks in computer networks [1]. Generally, malware is a term for all types of malicious program. It is the type of software that is used with the aim of attempting to breach the security policy of computer system or network with respect to Confidentiality, Integrity or Availability [2]. Authors in [3] have also argued that malware serves as a malicious software that has caused serious disruption to IT or digital assets globally as [4] further buttressed that malware have been widely used to attack personal and organization computer

systems as well as related devices. According to [5], a malware is a malicious applications and code that can cause damage and disrupt normal use of devices. [6] mentioned that the most common types of malware include viruses, trojan horses, ransomware and many others. Therefore, researchers in [7] have traced the origin of malware to 1980s when some investigators came up with self-replicating computer programs. There are different types of malware in Windows Operating System. One of the varieties that malware appear is in Portable Executable Files format, VB scripts, Java Scripts, Macros in Windows files, as well as exploits in files [7], Moreover, [8] have argued that malware keep changing in their characteristics and they are popular hazards in computer networks and cyber space. Authors in [9] as well as the researchers in [10] have explained the powerful characteristics of evolving malware that have the capability to change their forms over time and can evade detection schemes. Thus, these kinds of malware inflict serious harm on computers and networks in multiple ways [11] and there is a need to step up detection models that can match the sophisticated security threats of malware. Investigating and identifying malware evidence using machine learning that is based of portable executable files characteristics is a promising approach since [7] pointed out that in the cyber space, the malware with executable files are about 805 of the total types available. Irrespective of the kind of operating system that is installed on a device, malware can infect it and cause unwanted and nasty experience for the computer user. For this reason, it can be argued that having an effective model that can be used for the identification of malware evidence is a step in the right direction. Despite the promises of the ML methods in the identification of malware evidence, some of the ML approaches in literature have poor detection rates which can be as a result of the size and nature of the patterns in the datasets used.

The dataset chosen in this study contains Portable Executable files that represent the presence of malware samples. According to [12], the Portable Executable (PE) format is a file format for executables, object code, Dynamic Link Libraries (DLLs) and others used in 32-bit and 64-bit versions of Windows operating systems, and in UEFI environments. This study used two ML algorithms for the identification of malware evidence in the chosen dataset. The approach is based on employing detailed exploratory analyses, cleaning and feature sub-set selection methods on the raw version of the CLaMP dataset. The results of the models built in this study are then compared based on accuracy, precision, recall, F1-score and Area Under the Curve (AUC) as metrics.

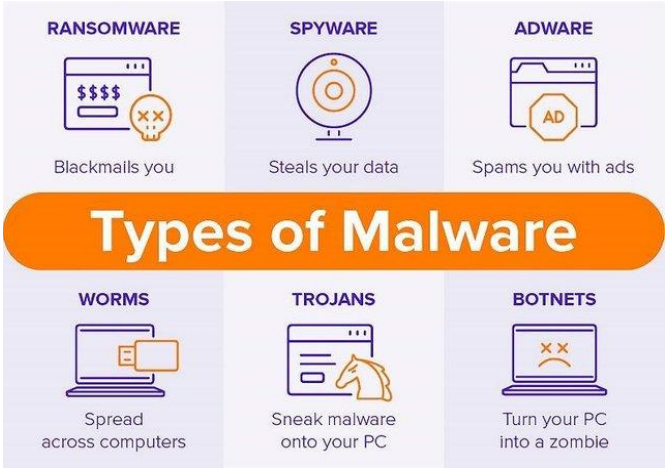


Figure 1: Types of Malware [6]

II. RELATED WORK

[13] compared seven machine learning and deep learning algorithms for the detection of malware by using the extracted byte, opcode, and section codes. In this research, we aim to classify malware in nine different malware families correctly.[13] argued that the deep learning method achieved better results in the classification of malware. [14] proposed malware classification models using six different learning algorithms. The authors made use of CIC-MalMem-2022 dataset for building the models. The six models were cross-validated by 10-fold cross validation. Good results were reported for the four metrics used. [15] experimented on how Tree-Based Learning Models can be used for the classification of Botnet as threat in network and internet space. The study made use of three different ensemble algorithms namely: Extra Trees, Random Forest and Adaboost. Authors proved that RF-based model achieved the best classification results.

Similarly, [16] proposed an XGBoost machine learning approach improved the identification of network intrusions, the authors pointed out that the approach is very promising for the identification of network intrusions in the chosen dataset based on the performance metrics used. [17] came up with an approach that used machine learning technique for the detection of malware in PE files in the chosen EMBER dataset. The authors pointed out that the approach achieved promising results. Ember is labeled benchmark dataset for training machine learning models for the detection of malicious Windows portable executable files.

[18] proposed an approach for the analysis of the malware samples as well as the identification of their malicious activities. [19] came up with shallow and deep learning machine methods for the classification of malware. Specifically, the author used the following learning algorithms for the malware classification: Support Vector Machine (SVM) and Gaussian Naive Bayes, Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN). Promising results were recorded with each of the algorithms. Recurrent

Neural networks was reported to be best approach that recorded the highest accuracy. [20] built machine learning model for the detection of fileless cryptocurrency mining malware. The proposed method was used for classifying conventional malware and cryptocurrency mining malware. The researchers used the dataset named EMBER dataset in the study. The authors mentioned that the approach is very effective compared to similar studies. [21] built various machine-learning algorithms such as Decision Tree, Random Forest,
[19] KNN, Logistic Regression, Linear Discriminant Analysis and Naive Bayes for the classification of malware. The researchers made us of CLaMP malware dataset.

[22] proposed machine learning model for the classification of malware in Android environment. The authors argued that the technique proposed was very promising in the classification of varieties of Android malware. The authors presented two machine learning aided approaches for static analysis of the mobile applications: one based on permissions, while the other based on source code analysis that utilizes a bag of words representation model. It was argued that the approach achieved code based classification with F-score of 95.1% compared to similar studies. [23] built various machine-learning algorithms such as Decision Tree, Random Forest, KNN, Logistic Regression, Linear Discriminant Analysis and Naive Bayes were adopted in the classification of malware. The researchers made use of CLaMP malware dataset.

III. METHODOLOGY

The methodology applied in this study involved carrying out a comprehensive exploratory analyses on the dataset. Thereafter, use the appropriate data cleaning methods before the classification of malware evidence in the dataset can be carried out. The two selected learning algorithms are then used for the classification of malware evidence in both the unprocessed and processed dataset and their results were compared.

A. Problem Formulation

Given a binary class malware dataset, the target is to effectively classify the presence or otherwise of malware in the chosen dataset based on Naive Bayes and Logistic Regression algorithms. The two learning algorithms are evaluated based on how they are able to effectively identify the presence malware types with PE headers with or without some data wrangling and feature selection approaches.

B. Description of Dataset

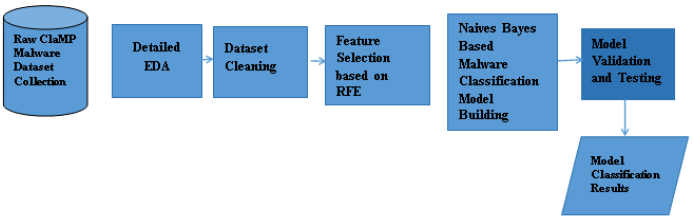
The CLaMP dataset used in this paper was obtained from Mendeley Data, V1, doi: 10.17632/xvyv59vwvz.1 [24] .The name of the dataset is an acronym of Classification of Malware with PE Headers. It is a malware dataset that contains portable executable files for malware detection. The full dataset contains raw and integrated data in both csv and rfff formats. This study made use of raw version of the dataset in its csv format. There are 5184 samples (rows) and x 56 attributes (columns), the last being the target class.

The input features in the dataset include*:*
[e_magic, e_cblp, e_cp, e_crlc, e_cparhdr, e_minalloc, e_maxalloc, e_ss, e_sp, e_csum, e_ip, e_cs, e_lfarlc, e_ovno, e_res, e_oemid, e_oeminfo, e_res2, e_lfanew, Machine, NumberOfSections, CreationYear, PointerToSymbolTable, NumberOfSymbols, SizeOfOptionalHeader, Characteristics, Magic, MajorLinkerVersion, MinorLinkerVersion, SizeOfCode, SizeOfInitializedData, SizeOfUninitializedData, AddressOfEntryPoint, BaseOfCode, BaseOfData, ImageBase, SectionAlignment, FileAlignment, MajorOperatingSystemVersion, MinorOperatingSystemVersion, MajorImageVersion, MinorImageVersion, MajorSubsystemVersion, MinorSubsystemVersion, SizeOfImage, SizeOfHeaders, CheckSum, Subsystem, DllCharacteristics, SizeOfStackReserve, SizeOfStackCommit, SizeOfHeapReserve, SizeOfHeapCommit, LoaderFlags, NumberOfRvaAndSizes] while the target variable is named **class**.

C. Architectural Description of the ML Method in the Study

The summary of the flow of activities in the ML methods used in this study is as capture in figures 2 and 3.

Figure 2: Architecture of A Naive Bayes-based Model for Malware Evidence



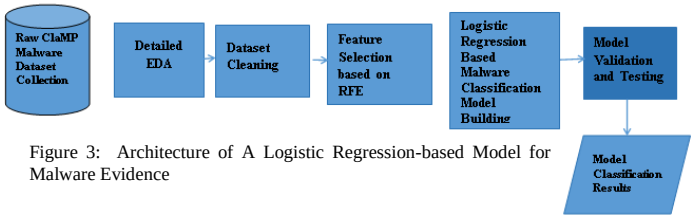
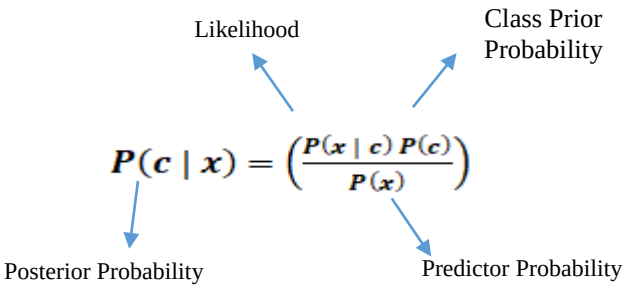


Figure 3: Architecture of A Logistic Regression-based Model for Malware Evidence

D. Classification Algorithms Used

- Naïve Bayes Algorithm
  Naive Bayes (NB) algorithm is a classification algorithm that is based on Bayes' Theorem with an assumption that all the features that predicts the target value are independent of each other. It calculates the probability of each class and then pick the one with the highest probability. Given a features vector $X=(x_1,x_2,\ldots,x_n)$ and a class variable y, Bayes Theorem calculates the posterior probability $P(y \mid X)$ from the likelihood $P(X \mid y)$ and prior probabilities $P(y)$,$P(X)$ for the classification task.

Arithmetic Representation of Naive Bayes Algorithm



$$P(c \mid x) = P(x_1 \mid c) \times P(x_2 \mid c) \times \ldots \times P(x_n \mid c) \times P(c) \quad (1)$$

The working principle of the NB algorithm was useful in the malware classification in this study

- Logistic Regression Algorithm
  Logistic regression (LR) algorithm is a machine learning method for classification tasks. Logistic regression is employed to predict the probability that a given input belongs to a particular class. In binary classification, we often label the two classes as 0 and 1. The LR model can be expressed mathematically as shown in equation 2.

Arithmetic Representation of Logistic Regression Algorithm

$$P(y = 1 \mid x) = \frac{1}{1 - e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}} \quad (2)$$

The logistic function is known as the sigmoid function and is very useful in logistic regression as it is used to map any real number into the range [0, 1].

The sigmoid function is mathematically obtained from

$$y = \frac{1}{1 + e^{-z}} \quad (3)$$

The working principle of LR algorithm was also useful in the malware classification in this study.

E. Metrics for Malware Classification Evaluation

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F1 score} = 2 \times \frac{\text{Precision x Recall}}{\text{Precision + Recall}} \quad (7)$$

$$\text{AUC} \approx \sum_{i=1}^{n-1} \left(\frac{TPR_{i+1} + TPR_i}{2}\right) \times (FPR_i + 1 - FPR_i) \quad (8)$$

Where:
TP = True Positive; TN = True Negative;
FP = False Positive, FN = False Negative

IV. RESULTS AND DISCUSSION

A. Results Experimental Set Up

All experimentation are carried out in Python environment in a Windows-based System of Corei3 Processor, 1TB Hard Disk Drive (HDD), 8GB RAM. The Python environment is an IDE named Spyder with various libraries such as panda, numpy, seaborn, Matplotlib and a few other packages. For instance, for the Recursive Feature Elimination method applied for the feature subset selection in the study, Boruta package was made use of.

- Results of Exploratory Data Analysis (EDA)

Table 1: Features and Samples in the Raw CLaMP Malware Dataset before pre-processing

	e_magic	e_cblp	e_cp ...	LoaderFlags	NumberOfRvaAndSizes	class
0	23117	144	3 ...	0	16	0
1	23117	144	3 ...	0	16	0
2	23117	144	3 ...	0	16	0
3	23117	144	3 ...	0	16	0
4	23117	144	3 ...	0	16	0
...	...	...	...	...	...	...
5179	23117	144	3 ...	0	16	1
5180	23117	80	2 ...	0	16	1
5181	23117	144	3 ...	0	16	1
5182	23117	144	3 ...	0	16	1
5183	23117	144	3 ...	0	16	1

Table 1 shows the features and samples in the chosen CLaMP dataset.

Table 2: Summary Statistics of the Malware Dataset

	e_magic	e_cblp ...	NumberOfRvaAndSizes	class
count	5184.0	5184.000000...	5184.000000	5184.000000
mean	23117.0	145.966435 ...	15.963927	0.517554
std	0.0	512.429759 ...	0.749333	0.499740
min	23117.0	0.000000 ...	0.000000	0.000000
25%	23117.0	144.000000 ...	16.000000	0.000000
50%	23117.0	144.000000 ...	16.000000	1.000000
75%	23117.0	144.000000 ...	16.000000	1.000000
max	23117.0	37008.000000	16.000000	1.000000

Table 2 is used to show the statistical summary of the features in the CLaMP dataset.

B. Data Types of the Features

From the exploratory analysis of the dataset, it was also observed that the dataset has 55 input attributes and one target

class. The data types in the dataset include two features with floating point values (float64) while fifty four features are on integer types (int64).

C. Handling the missing values in the dataset

The detailed EDA carried out in this study also revealed that there are two columns that are completely empty (missing) in the dataset. The attributes are e_res and e_res2. The method used for handling the missing values is to delete the two columns completely. This technique was used since there is no any values at all that can be used for any kind of imputation for handling missing values as supported in literature.

D. Dataframe of Duplicate Values in the Dataset

There are duplicate rows (samples) in the dataset. From the EDA carried out, it was revealed that there are 624 duplicate values. The duplicated values (rows) are removed through deletion. For that reason, the original dimension of the dataset which is (5184, 55) has been reduced to (4560, 53) after pre-processing. The last column (attribute) is the target variable. The dataframe of the rows identified as duplicated values are as shown in table 3.

Table 3: Dataframe of duplicated rows found in the raw CLaMP Dataset

	e_magic	e_cblp	e_cp	...	LoaderFlags	NumberOfRvaAndSizes	class
541	23117	144	3	...	0	16	0
779	23117	80	2	...	0	16	0
780	23117	144	3	...	0	16	0
837	23117	80	2	...	0	16	0
856	23117	144	3	...	0	16	0
...	...	...	...	...	...	...	...
5165	23117	80	3	...	0	16	1
5170	23117	144	3	...	0	16	1
5173	23117	80	2	...	0	16	1
5176	23117	144	3	...	0	16	1
5183	23117	144	3	...	0	16	1

Table 3 shows the duplicated rows that are found in the chosen dataset used for the study.

E. Classes in the Dataset

The number of classes in the dataset is 2. Thus, the dataset can be used for binary classification: malware and non-malware traffic. It was also revealed that the dataset did not suffer from extreme class imbalance. Thus, this study did not consider class imbalance as an issue.

F. Result of the Feature selection method used

In the dataset, each sample is a dimensional vector and the set of feature is regarded as feature vector. Based on the EDA of the dataset, there are 5184 rows (instances) and 56 attributes (55 inputs and one target variable) that are originally in the CLAMP data**.** Due to the fact that the dataset has slightly large number of features, there is a need to select promising attributes. Feature selection methods are used in machine learning (ML) tasks for the purpose of reducing the model complexity, and improve model performance [25], [26], [27] The feature selection technique used in this study is Recursive Feature Elimination (RFE) method. It is a type of Wrapper feature selection method. The Feature selection was carried out so as to determine the most important features that contribute to malware classification, After the data cleaning and the dimension of the dataset was reduced to (4560, 53). With the recursive feature selection technique, the performances of the Naive Bayes and Logistic Regression models were measured with different subsets of features and incidentally the features that produced the highest performances across the five metrics were used.

G. Malware Classification with the raw CLaMP Dataset

The classification models from the two selected algorithms were built by putting certain measures into consideration. The two chosen learning algorithms used in this study work based on probabilities to establish the existence of malware or not in the processed dataset. The EDA carried out provided a good ground on how the dataset was cleaned. Also, the RFE attribute selection approach used enable us to select different features severally until optimal results were obtained for all the metrics. In the two scenarios, a train-test split ratios of 90:10 were used. The train-test split ratios used were varied. For each of presentation, the conditions under which the two scenarios work include when the models were built from the raw data without pre-processing and sub-set feature selection as well as when they were built from the data with pre-processing and sub-set feature selection. The Performance Metrics used for evaluation include: Accuracy, Precision, Recall, F1-Score and AUC Score. The results are captured in tables 4 and 5.

- **Scenario 1**

This case is used to analyse the classification of malware evidence in the raw dataset without carrying out data cleaning and attribute selection on the dataset despite the many issues obtained from the exploratory data analyses (EDAs). The malware classification results of the selected NB and LR algorithms in the first scenario are as shown in table 4.

Table 4: Malware Classification Result based on Scenario 1

Learning Algorithm/Metrics	**Naive Bayes**	**Logistic Regression**
Accuracy (%)	52.4590	52.6012
Precision	0.5242	0.5260
Recall	0.9972	0.9970
F1-Score	0.6874	0.6894
AUC	0.5011	0.5000

- **Scenario 2**

This scenario is where the ML-based models are used to analyse the classification of malware evidence in the raw CLaMP dataset by carrying out data cleaning of the dataset, perform feature subset selection based on the conclusions drawn from the exploratory data analyses (EDAs). The results of the NB and LR algorithms under this scenario are as shown in table 5.

Table 5: Malware Classification Result based on Scenario 2

Learning Algorithm/Metrics	**Naive Bayes**	**Logistic Regression**
Accuracy (%)	99.9930	99.8700
Precision	1.0000	1.0000
Recall	0.9990	0.9980
F1-Score	0.9990	0.9982
AUC	1.0000	0.9990

H. Discussions

The defectiveness of signature-based detection approaches have been reported which gave room machine learning-based malware identification techniques to be more popular, in recent times. Despite the promises of the ML methods, some of the ML approaches reported in literature have their own limitations as they recorded high positive rate. This study first of all emphasised the need to study the patterns in the CLaMP dataset. Based on the EDA carried out on the dataset, it was observed that the target class is binary in nature. The comprehensive exploratory analyses of the dataset also revealed better understanding of the data distributions for malware identification. Thus, this paper focuses on solving a binary class problem of identifying whether there is malware evidence or not in the dataset. In the dataset, there are some issues that were addressed before feeding the dataset into the learning algorithms. Thereafter, Naive Bayes (NB) and Logistic Regression (LR) algorithms were used for building the malware identification model both without and with data cleaning and feature selection. The hyper parameters of the algorithms were varied and the models were tested and validated severally before optimal performances were arrived at. Tables 4 and 5 are used to capture the results of the two models in respect of malware identification. The results of the models were compared based on the selected metrics, namely: accuracy, precision, recall, f1-score and Area Under the Curve (AUC).In scenario 1, the dataset was not pre-processed and all the attributes were used for the model building. Poor results were obtained by both models but NB based malware identification model except in recall. Also, NB-based model slightly performed better than LR in all the metrics. It was also discovered that both NB and LR models performed well in scenario 2 when the dataset was pre-processed and promising features were selected compared to scenario 1. Recall that in ML, the closer the AUC is to 1, the better the model. The general experimental result further established that the EDA, data pre-processing, feature selection and hyper parameter tuning helped achieve good malware classification results in the study. Results obtained in this study outperformed the ones in similar studies.

V. CONCLUSION

This study focuses on investigating how exploratory data analyses, data cleaning and feature selection approach that is based on RFE can influence building of effective machine learning models for malware classification task. The study employed some innovative approaches for the exploratory analyses of the chosen dataset so as to reveal the real distributions in the dataset. Thus, it leads the research further for the choice of appropriate methods that can be used for the treating the issues in the dataset prior to using it for building malware identification models. From the analyses carried out, it was found out that the data distributions have effect on how the learning algorithms behave in two different scenarios. In the first scenario, the raw dataset was used for the training and testing purposes by feeding the two supervised learning algorithms with unprocessed data features. Improved malware classification results were recorded in the second scenario by varying the approaches. For instance, promising performances of the two models were achieved by using the appropriate techniques to address some of the anomalies or limitations in the dataset before building the NB and LR-based malware identification models in the second scenario. In all Naive Bayes-based malware identification model slightly performed better than the Logistic Regression based model in both scenarios one and two. The general conclusion is that the detailed exploratory analyses of the dataset which reflected the many issues that were addressed in the second scenario were helpful in building very promising malware identification models using the two learning algorithms. The study concluded that the EDA, data cleaning and feature subset selection enabled the researchers to achieve promising results in the malware identification tasks.

ACKNOWLEDGMENT

Authors acknowledge the efforts of the anonymous reviewers who reviewed the paper and contribute to make it accepted.

REFERENCES

[1] O. M. Ayinla, A. M. Oyelakin, and J. O. Olomu, "A Comprehensive Review On Machine Learning Techniques For The Identification of Ransomware Attacks In Networks," *LAUTECH Journal of Computing and Informatics (LAUJCI)*, vol. 4, no. 1, Mar. 2024.

[2] R. Sharp, "An Introduction to Malware," 2017. [Online]. Available:https://backend.orbit.dtu.dk/ws/portalfiles/portal/139067614/malware.pdf. [Accessed: May 23, 2024].

[3] N. Kumar, S. Mukhopadhay, M. Gupta, A. Handa, S. K. Shukla (2019). "Malware Classification using Early Stage Behavioural Analysis",

Conference: *2019 14th Asia Joint Conference on Information Security* (AsiaJCIS), DOI: 10.1109/AsiaJCIS.2019.00-10

[4] M. A. H. Saeed, "Malware in Computer Systems: Problems and Solutions," *IJID International Journal on Informatics for Development*, vol. 9, no. 1, pp. 1-8, 2020. doi: 10.14421/ijid.2020.09101.

[5] Microsoft Windows Security, "Understanding malware & other threats - Windows security," *Microsoft Docs*, 2019. [Online]. Available: https://learn.microsoft.com/sr-Latn-RS/defender-endpoint/malware/understanding-malware

[6] Avast, "What is Malware & How Does it Work? Malware Definition," *Avast*, 2019.

[7] Quick Heal R & D Lab, "Introduction to Malware and Malware Analysis," pp. 1-8, 2014. [Online].Available:http://dlupdate.quickheal.com/documents/technical_papers/introduction_to_malware_and_malware_analysis.pdf. [Accessed: May 23, 2024].

[8] [8] D. Ucci, L. Aniello, and R. Baldoni, "Survey of Machine Learning Techniques for Malware Analysis," *Computers & Security*, vol. 81, pp. 123–147, 2019. doi: 10.1016/j.cose.2018.11.007.

[9] [9] A. M. Oyelakin and R. G. Jimoh, "A Review on the Identification Techniques for Detection-Evasive Botnet Malware," in *Proceedings of Nigeria Computer Society (NCS)*, July 2019 International Conference of NCS, Gombe, Gombe State, Nigeria, 2019.

[10] [10] A. Bulazel and B. Yener, "Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium on - ROOTS," pp. 1-21, 2017.

[11] Kaspersky Labs, "What is malware and how to defend against it?" 2017. [Online]. Available: http://usa.kaspersky.com/internet-security center/internet-safety/what-is-malware-and-how-to-protect-against it#.WJZS9xt942x.

[12] Trend Micro, "Portable executable (PE)," 2022. [Online]. Available: www.trendmicro.com.

[13] B. Bokolo, R. Jinad, and Q. Liu, "A Comparison Study to Detect Malware using Deep Learning and Machine Learning Techniques," in *2023 IEEE 6th International Conference on Big Data and Artificial Intelligence (BDAI)*, pp. 1-6, 2023. doi: 0.1109/BDAI59165.2023.10256957.

[14] S. Kumar, S. Singh, S. Kumar, and K. Verma, "Malware Classification Using Machine Learning Models," in *International Conference on Machine Learning and Data Engineering (ICMLDE 2023)*, *Procedia Computer Science*, vol. 235, pp. 1419–1428, 2024.

[15] A. M. Oyelakin and R. G. Jimoh, "Tree-Based Learning Models for Botnet Malware Classification in Real Life Sub-Sample Dataset," *Innovative Computing Review*, vol. 3, no. 2, pp. 1-13, Dec. 2023.

[16] A. M. Oyelakin, M. B. Akanbi, T. S. Ogundele, A. O. Akanni, M. D. Gbolagade, M. D. Rilwan, and M. A. Jibrin, "A Machine Learning Approach for the Identification of Network Intrusions Based on Ensemble XGBOOST Classifier,"*Indonesian Journal of Data and Science*. [Online]. Available: https://jurnal.yoctobrain.org/index.php/ijodas/article/view/88/167.

[17] C. Connors and D. Sarkar, "Machine Learning for Detecting Malware in PE Files," *arXiv*, Dec. 2022. [Online]. Available: arxiv:2212.13988v1 [cs.CR].

[18] M. R. Zaharin and S. M. Shariff, "Malware Classification based on System Call," in *Advances in Visual Informatics*, *7th International Visual Informatics Conference (IVIC 2021)*, Kajang, Malaysia, Nov. 23-25, 2021, pp. 387-398, ACM Digital Library. doi: 10.1007/978-3-030-90562-5_34.

[19] S. Dilhara, "Classification of Malware using Machine Learning and Deep Learning Techniques," *International Journal of Computer Applications*, vol. 183, no. 32, pp. 12-17, 2021. doi: 10.5120/ijca2021921708.

[20] W. Handaya, M. N. Yusoff, and A. Jantan, "Machine learning approach for detection of fileless cryptocurrency mining malware," *Journal of Physics: Conference Series*, vol. 1450, no. 1, p. 012075, 2020. doi: 10.1088/1742-6596/1450/1/012075.

[21] A. Kumar, K. S. Kuppusamy, and G. Aghila, "A learning model to detect maliciousness of portable executable using integrated feature set," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 2, pp. 252-265, 2019. doi: 10.1016/j.jksuci.2017.01.003.

[22] N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, "Machine learning aided Android malware classification," *Computers & Electrical Engineering*, vol. 61, pp. 266-274, 2017. doi: 10.1016/j.compeleceng.2017.02.013.

[23] A. Kumar, K. S. Kuppusamy, and A. Gnanasekaran, "A learning model to detect maliciousness of portable executable using integrated feature set," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 2, 2017. doi: 10.1016/j.jksuci.2017.01.003.

[24] A. Kumar, "ClaMP (Classification of Malware with PE headers)," *Mendeley Data*, V1, 2020. doi: 10.17632/xvyv59vwvz.1.

[25] L. Ladha and T. Deepa, "Feature Selection Methods and Algorithms," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, no. 5, pp. 1787–1797, 2011.

[26] A. M. Oyelakin and R. G. Jimoh, "A Survey of Feature Extraction and Feature Selection Techniques Used in Machine Learning-Based Botnet Detection Schemes," *VAWKUM Transactions on Computer Sciences*, vol. 9, pp. 1-7, 2021. [Online]. Available: https://vfast.org/journals/index.php/VTCS/article/view/604/658.

[27] Y. Lyu, Y. Feng, and K. Sakurai, "A Survey on Feature Selection Techniques Based on Filtering Methods for Cyber Attack Detection," *Information*, vol. 14, p. 191, 2023. doi: 10.3390/info14030191.