IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 24, Issue 1, Ser. III (Jan. –Feb. 2022), PP 23-30
www.iosrjournals.org

A Machine Learning Model for Malware Detection Using Recursive Feature Elimination (RFE) For Feature Selection and Ensemble Technique

Baffa Sani Mahmoud
Department of Computer Science
Sule Lamido Universty, Kafin Hausa, Nigeria

Prof. Ahmad Baita Garko
Department of Computer Science
Federal University Dutse,Jigawa, Nigeria.

Abstract- *In recent years, almost every member of the society has been using Internet for their daily life. This is because it has become an integral part of our life. But, as the technology is continually advancing, the cyber security threats such as spamming and malwares, are swarming cyberspace exploiting vulnerabilities, making it harder for computer users to keep their data safe. To be able to keep people and files safe, an antivirus programs were created to aid in malware detection and prevention. While the performance of antivirus is commendable, they suffer from two major drawbacks, their inability to detect new malwares whose signatures are not available in the virus definition at the time, and its inability to detect zero day attack. Recently, researchers proposed machine learning models that are capable of detecting zero day attacks, but these models are also vulnerable to an adversarial attacks. To overcome this problem, an ensemble boosting model for malware detection is proposed in this research, and also the role of Recursive Feature Elimination (RFE) algorithm on PE header files dataset is investigated. The proposed model achieved a state of the art accuracy of 99%, precision of 98% and Recall of 97%. The model also outperformed other existing model when their performances were compared.*

Keyword: *Ensemble, Machine learning, Malware detection, Recursive Feature Elimination (RFE)*

Date of Submission: 10-02-2022 Date of Acceptance: 25-02-2022

I. Introduction

In recent years, almost every member of the society has been using Internet for their daily life. This is because it is an integral part of our life nowadays, it is becoming impossible to do anything without the Internet including social interactions, online banking, health related issues, and marketing [1].

As technology is continually advancing, so the cyber security threats such as spamming and Malware, are always evolving and becoming more dangerous, making it harder for computer users to keep their data safe. Malware is any software intentionally designed to cause damage to a computer, server, client, or computer network. Malware is the collective name for a number of malicious software variants, including viruses, ransom ware and spyware, Trojan horse and so on [2].

Today we are completely living in computing world where most of the people are connected to the internet as such keeping their personal data and information safe is becoming a tedious task. Personal data is very much available to public database, these data are vulnerable to threats in which malware developers or hackers can exploits. Therefore it is important for the security companies, government agencies and social media platform owners to detect these malicious program for their privacy and safety of the contents as well as users. In line with extreme growth of the internet, variety of new malware are produced on daily basis, making it difficult to detect and analyze manually [3]. Currently, the known malwares are created mainly for, stealing sensitive data, fraudulent transactions, social engineering, or for a ransom. There are so many ways to get infected with these malwares on line, the most common is downloading files from suspicious sites. .

To be able to keep people and files safe in cyberspace, malware detection are used to constantly scan and detect malwares in our systems Malware detection is the process of determining whether a given program has malicious intent or not. While malware classification is the process of classifying malware samples based on shared characteristics with previously analyzed samples. E.g. Strings and binary codes.

The earliest technique used in detecting malware is signature-based, majority of available commercial antivirus solution used this method, while the method is quite effective, but it suffers from one major drawback,

its inability to detect unknown and new generation malwares [10]. To overcome this drawback, researchers proposed the use of machine learning techniques and deep neural networks [11]. This have proven to be more effective in detecting zero-day attacks which are beyond the capability of commercial off the shelve antivirus software.

Currently, so many researches have been made on Malware detection and classification using machine learning algorithms, but still there is more to investigate. As the researchers investigate and provide best ways to detect malwares, so does the hackers, and working tirelessly to undermine such methods.

In the existing models, adversarial attacks were not considered. Adversarial attacks are special attacks that hackers used to create malicious program to attack the machine learning detection models, either by corrupting the dataset or make the model misclassify malwares, allowing malwares to pass through the model undetected.

The rise of these attacks which target the detection models calls for other model that can be able to mitigate such kind of attacks. Fortunately an ensemble machine learning model is proven to overcome such attack as mentioned in [12].

Therefore, in this research an ensemble boosting model for malware detection is proposed, and also the role of Recursive Feature Elimination (RFE) algorithm in malware detection using PE header files dataset is investigated. The PE header dataset [8] is very large with about 20, 000 instances and over fifty (50) features. This high dimensionality affects model performance causing longer training time and even lower accuracy. The proposed model used RFE techniques to select bet features that is enough to provide great model performance with shorter training time, which was not the case in the previous researches.

The rest of the paper is organized as follows, Section II discussed related works find in literature in relation to malware detection and feature selection. Section III provides the overall methodology employed in this research starting from method of data collection, data preprocessing, model definition as well as model building. Section IV presents the results of the experiments conducted over the cause of this research, and finally Section V highlight the conclusion and future research direction.

II. Related works

Machine learning: is a branch of artificial intelligence that allows computer systems to learn directly from examples data, and experience [6]. Machine learning is the science of getting computers to learn and act like humans do, and improve their learning over times. Various machine learning techniques are used for malware detection and classification, among others are Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest and many others [7].

Portable Executable (PE) files: are file formats for executable, DDLs, and object codes used in 32-bit and 64-bit versions of Windows. They contain many useful pieces of information for malware analysts, including imports, exports, time-date stamps, subsystems, sections, and resources.

Malware detection was done in the past variety of ways, starting from manual code scanning, to automatic scans, and used of antivirus programs. The techniques examine different file in order to detect the presence of malware or not. According to Aslan et al [1] Malware detection is the process of determining whether a given program has malicious intent or not. It's also the process of scanning the computer and files to detect malware.

Early researchers in the field of cyber security has further subdivided this detection process into two categories which are (1) signature based detection, (2) behavioral based detection. The signature based detection is usually used by antivirus software's. This detection method relies on already known file fingerprints, static strings or file metadata which are known to be malicious. The signature based detection works by scanning a program or a file collects the code and sends it to a cloud-based database. The database has a vast collection of virus codes. If the file code is found in the list, the database returns with a verdict that the file is malware. The major drawback of this technique is its inability to detect new malwares, which signature is not available in the antirust.

To overcome the drawback of signature based method researchers proposed the second category of using heuristics or behavioral method, in this method malwares are identified by inspecting programs behaviors and analyzing specific malware behaviors, specific behavioral patterns can be attributed to malware and build a model that can extract such behaviors from programs and identified malwares. The behavioral methods paved way for machine learning algorithms to be used in malware detection.

Machine learning (ML) algorithms were used widely used in malware detection and classification using different algorithms. The use of Machine Learning for threat detection is essential to counter the massive growth in malware. AV-Test, an independent research institute for IT security, claims it detects an astonishing 350,000 new malware samples every day. The company has calculated that over 972 million malware specimens are currently swarming the internet [4].

Machine learning in cyber security had gone further by enabling anti-malware software to learn which files are malicious and which are benign based on patterns learned. It also makes decisions about whether or not the analyzed file is malicious or benign.

Furthermore, Machine learning techniques have proved to be capable for identifying a zero-day attack which can be one of the most deadly types of malware threats. According Kamalakanta [5] it is a general believe by cyber security experts that the use of antimalware tools and systems powered by artificial intelligence, and machine learning will be the solution to modern malware attacks.

III. Methodology

This section introduced the methodology employed in this research work, it describe how the dataset was obtained and pre-processed. The section discussed the model formulation and techniques used to evaluate the model.

3.1Data Gathering

The dataset used in this research was obtained from kaggle.com [8], an open source dataset repository for data scientist, or an online community of data scientists and machine learning practitioners as it contained more than a dataset. It was release under CC by NC 3.0 license solely for research.

3.2 Dataset Description

The dataset is a made-up of several Portable Executable (PE) files that were recorded by from virusShare. The PE format is a data structure that encapsulates the information necessary for the Windows operating system loader to manage the wrapped executable code which includes several program features. Each executable program have these attribute before it can be executed by the operating system, the dataset contains files from both benign and malware program. The dataset have a total of 19,612 instances, each instance in a dataset have 78 features including program ID and a class which describe whether the instance is malware or benign. The dataset is distributed almost evenly, in the sense of classes between benign and malware instances. Fig.1 display a sample of raw dataset prior to any pre-processing operation.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Name	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	e_minallo	e_maxallc	e_ss	e_sp	e_csum	e_ip	e_cs	e_lfarlc	e_ovno	e_oemid	e_oeminf	e_lfanew	Machine	NumberO
2	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248	34404	6
3	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240	332	5
4	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256	332	6
5	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	128	332	7
6	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	128	332	7
7	VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256	332	8
8	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248	332	5
9	VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256	332	8
10	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256	332	4
11	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224	332	7
12	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224	332	7
13	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	264	332	4
14	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	264	332	6
15	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256	34404	8
16	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224	332	3
17	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248	332	5
18	VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248	332	3
19	VirusShar	23117	144	3	0	4	0	17744	0	332	1	29305	15462	19547	29295	267	6	12	332	1

Fig.1 sample dataset

3.3 Data Pre-Processing

The standard practice of machine learning requires dataset to be pre-processed before it will be used by the model. The reasons for dataset pre-processing are many, some of which is, removing repetition or redundant features, removing the non- numeric features, as the model works only on the numeric features, separating the features and target class and handling missing values.

In this dataset no redundant features were found, but the dataset contains string features, and irreverent features such as program ID which has to be removed before use. Another column that was also removed is target class, that is the class that indicates whether the instance is benign or malware, usually represented as binary 0 or 1.

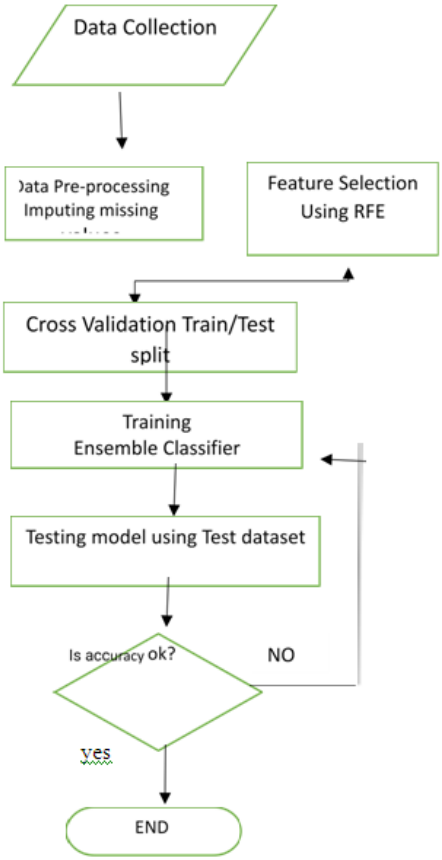
3.3.2 Handling Missing Values

Another interesting aspect in data pre-processing after data cleaning, is handling missing values, majority of machine learning algorithm do not work on data which contains missing values, that is why it is necessary to perform it. Some practice suggest that the missing values be replace with 0. As such loosing meaningful information that can be provided by this columns. Therefore, there is need for a systematic techniques to handle these missing values and replace them with meaningful information, as it often improved model performance when done with right technique.

Several methods have been used by researchers for missing value imputation in the dataset which occurs due to incorrect collection of data values. Some of the methods include mean, imputer method, k-means and median Senapti [9] .The malware dataset contains about twenty (20) missing values that belongs to many instance and several incorrect values. The missing values were computed using median imputer method

3.4 Model Definition

The proposed model can be described more formally using system flowchart Depicted in fig.2. The proposed model consists of the following modules: data collection, pre-processing stage which involves handling of missing data, dimensionality reduction, the training and testing of the machine learning models and lastly, performance analysis and comparison. Fig.2 depicts the proposed model framework. The preprocessing was done to handle 8missing data. Dimensionality reduction technique was employed to reduce the data features as the dataset contains an enormous features of 78 features which affect model performance.



The dataset was split into test and training sets using k-fold cross validation. An ensemble classifier was developed using Boosting approach to handle the classification aspect. It was proven in many researches that ensemble technique outperformed traditional machine learning algorithms in classification task, as such ensemble method was employed in this research. Performance metrics were used on the proposed model to evaluate the performance in terms of accuracy, precisions and recall. Implementation was carried out using the PYTHON programming language.

3.4.2 Recursive feature Elimination (RFE)

Experiment

Recursive Feature Elimination (RFE) is basically a backward selection of the features. The process begins with building a model using the entire set of features and computing an importance score for each feature. The least important feature are then removed, the model is re-built, and importance scores are computed again in a recursive manner. Users can specify the number of features they want to evaluate as well as each subset's size. Therefore, the subset size is a tuning parameter for RFE. The subset size that optimizes the performance criteria is used to select the features based on the importance rankings. The optimal subset is then used to train the final model.

The selected features formed what can be regarded as a reduced dataset, different in dimension with the original dataset. This reduced dataset was split into training and testing set, and performed K fold cross validation. K-Fold is a process of sharing training and testing data. The amount of training and testing data depends on the determination of partition K. In this research, the value of K used is 10. In K-Fold training data and testing data are carried out alternately. The training set will be pass to the ensemble classifier for training, the algorithm for the proposed model is represented below:

Proposed Model Algorithm

1. Given full training set (x_i, y_i)
2. Initialize: T, n_i
3. Repeat T times
4. For (i =1 to n)
5. Do
6. Proposed_Ensemble_ Model ← fit(xi, yi), F(n)
7. Compute F((n) importance scores))
8. While (importance score (ni)<threshold)
9. feature.Drop (n^i)
10. Go back to 5
11. Else
12. Back to 6
13. Return the remaining F (n^i)
14. End loop
15. End

Fig.3 Proposed Model Algorithm

Where n^i is vector containing all the features in the dataset. From the algorithm it can be inferred that RFE is repetitive process of fitting a model using different feature subset and dropping the least important.

3.5 Model Evaluation

The performance of the model was evaluated using standard metrics such as Accuracy, Precision, Recall, Confusion matrix and F1-score.

i-**Confusion matrix**: is a table that represents the ability of the model to classify labels correctly. It describes the performance parameters for the classifier as illustrated in

ii **Accuracy:** Total number of correctly classified instances divided by total numbers of instances as shown below:

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

iii- Precision: This describe the ability of the classifier to correctly identify the positive class. It can be calculate as:

$$\frac{TP}{(TP + FP)}$$

iv- Recall: The number of times the classifier predicted a negative class out of all the times the class is negative (benign) as seen below:

$$\frac{TP}{(TP + FN)}$$

v- Sensitivity is the metric that evaluates a model's ability to predict true positives of each available category.

$$\frac{TP}{(TP + FN)}$$

vi- Specificity is the metric that evaluates a model's ability to predict true negatives of each available category.

$$\frac{TN}{(TN + FP)}$$

3.6 Model Implementation

The proposed model was implemented using python programming language, it was built in Spyder, a python programming environment, which has a built in machine learning libraries such as Sci-Kit Learn.

The hardware resources used are:

1. Windows 10, 64bit PC.
2. 8GB RAM and 250GB SDD space
3. GPU Nvidia Geforce

4 **Results and discussion**

This chapter presents the result of the experiments carried out over the course of this research work. The chapter also discusses the results in details to highlight the significance of the results obtained in the research.

4.2 Data- Preprocessing

The first step in methodology is data preprocessing, where inconsistencies and errors found in the datasets were identified and mitigated. Such problems usually arose from data collection processes. Example of errors found in the dataset are missing values, wrong data and so on. The missing values were discovered and imputed using median imputer method.

4.3 Recursive Feature Elimination (RFE) Results

The malware dataset used in this research is a dataset categorized with high number features, the dimensions of the dataset is quite large. This undermines the performance of the model in terms of running time and accuracy. To overcome this problem, a features selection technique (RFE) was used to reduce the dataset dimensionality.

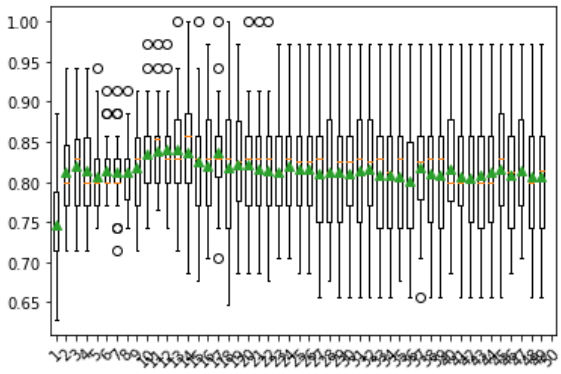


Fig.4 RFE results

The result provides us with only 15 features out of 78 that were selected by RFE algorithm. When fitted with single classifier, the selected features achieved all time high accuracy of 89%. This indicates that the 15 are enough to train the model for malware detection and achieved higher accuracy and faster inference time compared to using all the features in dataset.

4.4 Proposed Model Performance.

The proposed RFE Ensemble model was built by incorporating the RFE and ensemble model to utilized the advantages of RFE and ensemble techniques, thus improving the model performance in terms of accuracy and inference time, as it will be dealing with a lesser features than the on the original dataset with high dimension.

Performance of Ensemble Boosting Model

	precision	recall	f1-score	support
0	0.97	0.94	0.96	1004
1	0.98	0.99	0.99	2919
accuracy			0.98	3923
macro avg	0.98	0.97	0.97	3923
weighted avg	0.98	0.98	0.98	3923

Fig. 5 Proposed Model result

The proposed model in fig.5 was evaluated using the standard evaluation metrics such as Accuracy, precision recall and F1 score. The results shows that the proposed model was able to achieved a 98% accuracy when tested. This indicates a strong improvement on the traditional models as well the ensemble without feature selection. The result proves that combining feature selection and ensemble boosting do improve model accuracy.

The model was also evaluated on other performance measures to make it more robust, such as receiver operating characteristics (ROC) area under curve (AUC) score which is presented in the fig.6 below.

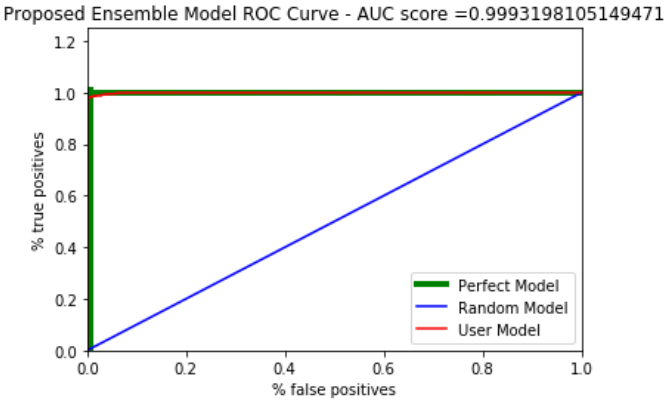


Fig. 6 Proposed Model Roc Curve

Receiver operating characteristics (ROC) Area under curve (AUC) score is a performance measure, used on models that solve classification problems at various threshold settings[13]. ROC represents a probability curve and AUC define the degree or measure of separability. This curve tells how much the model is capable of distinguishing between classes. High AUC score means that the model is good at predicting each class correctly. We can say that, the higher the AUC, the better the model is at distinguishing between malware and benign.

4.5 Performance Evaluation

The model performance was compared with the traditional machine learning classifiers to justify its performance. From the Table 5. It is clear that the proposed model outperformed all the compared traditional machine learning algorithms. The results of traditional classifiers displayed in Table 2 was gotten from the preliminary experiment carried out in the research. The ROC curve of the classifier is displayed in fig7 below.

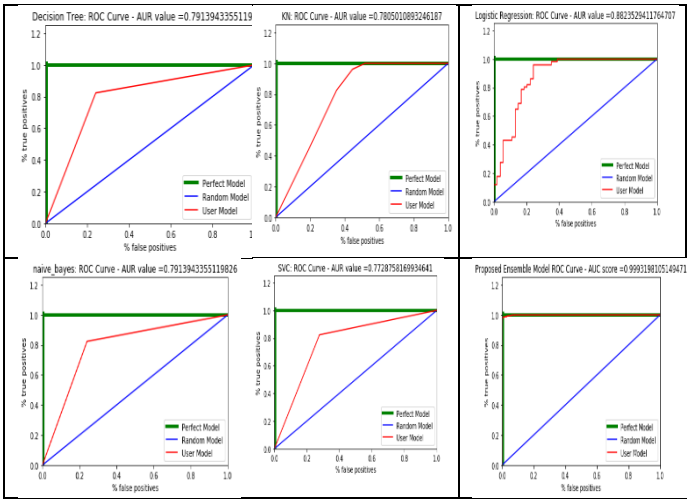


Fig. 7: proposed model ROC vs other models

In order to evaluate the performance of the proposed model with other existing models, a performance comparison was carried out between the proposed model and other existing models proposed by other researchers. This will give the model a generalization power on existing works and can serve as benchmark work as per as malware classification is concern using PE dataset.

Table 6: Proposed Model vs Other Existing Models

Models	AUC score	F1 Score	Precision	Recall
Ensemble-Minimum feature set model	99.8%	-	-	-
Efficient approach for malware detection	95.5	95.5	95.7	95.4
A learning model based on integrated feature set	94.9	94.9	95.5	94.4
Proposed model	99.9%	98%	98%	97%

IV. Conclusion and future work

From the results discussed in section 4, the performance of the proposed model is commendable, even when compared with other existing models, the proposed model outperformed existing models with significant increment in terms of accuracy. It shows that the proposed model is better than the existing malware detection models found in the literatures, also the model can be able to withstand adversarial attacks due to the ensemble technique employed. The research also proves the effectiveness of features selection on dataset with high dimensions as well as effect of using ensemble boosting technique. It is recommended that in future other feature selection techniques be investigated, to compare their performances and obtained the optimal technique.

References

[1]. Ö. A. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," in IEEE Access, vol. 8, pp. 6249-6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[2]. "What is Malware?," *Forcepoint*, Aug. 12, 2018. https://www.forcepoint.com/cyber-edu/malware#:~:text=Malware%20is%20the%20collective%20name,unauthorized%20access%20to%20a%20network. (accessed Jan. 02, 2022).

[3]. "Machine Learning and Images for Malware Detection and Classification Project: MaLiC Konstantinos Kosmidis," 2016. Accessed: Jan. 02, 2022. [Online]. Available: https://repository.ihu.edu.gr/xmlui/bitstream/handle/11544/15225/Machine%20Learning%20and%20Images%20for%20Malware%20Detection%20and%20Classification-MaLiC.pdf?sequence=1.

[4]. P. Lipman, "Machine Learning's Vital Role in Malware Detection," *AiThority*, Dec. 17, 2019. https://aithority.com/guest-authors/machine-learnings-vital-role-in-malware-detection/ (accessed Jan. 02, 2022).

[5]. Kamalakanta S., Rahul K., Lingaraj S., Padmalochan B. and Prashanta K. P (2019) A Novel Machine Learning Based Malware Detection and Classification Framework,

[6]. Vansh Jatana, "Machine Learning Algorithms," *ResearchGate*, 2019. https://www.academia.edu/40336766/Machine_Learning_Algorithms (accessed Jan. 03, 2022).

[7]. HarshaLatha, P., Mohanasundaram, R. (2020) Classification of Malware Detection Using Machine Learning Algorithms: A Survey, international journal of scientific & technology research volume 9, issue 02, February, 2020.

[8]. mauricio, "Benign & Malicious PE Files," *Kaggle.com*, 2018. https://www.kaggle.com/amauricio/pe-files-malwares?select=dataset_test.csv (accessed Dec. 10, 2021).

[9]. Senapti, R., Shaw, K., Mishra, S., & Mishra, D. (2012). A novel approach for missing value imputation and classification of microarray dataset. *Procedia engineering*, *38*, 1067-1071.

[10]. Jareth, "The pros, cons and limitations of AI and machine learning in antivirus software - Emsisoft | Security Blog," *Emsisoft | Security Blog*, Mar. 19, 2020. https://blog.emsisoft.com/en/35668/the-pros-cons-and-limitations-of-ai-and-machine-learning-in-antivirus-software/ (accessed Jan. 08, 2022).

[11]. A. Musa and F. Aliyu, "Performance Evaluation of Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF)," 2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf), 2019, pp. 1-5, doi: 10.1109/NigeriaComputConf45974.2019.8949669.

[12]. F. Abri, S. Siami-Namini, M. A. Khanghah, F. M. Soltani and A. S. Namin, "Can Machine/Deep Learning Classifiers Detect Zero-Day Malware with High Accuracy?," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3252-3259, doi: 10.1109/BigData47090.2019.9006514.

[13]. Sarang Narkhede, "Understanding AUC - ROC Curve - Towards Data Science," *Medium*, Jun. 26, 2018. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (accessed Dec. 07, 2021).

[14]. J. Shreffler and M. R. Huecker, "Diagnostic Testing Accuracy: Sensitivity, Specificity, Predictive Values and Likelihood Ratios," *Nih.gov*, Mar. 03, 2021. https://www.ncbi.nlm.nih.gov/books/NBK557491/ (accessed Dec. 07, 2021).

[15]. Souri, A., & Hosseini, R., Jan 2018. A state-of-the-art survey of malware detection approaches using data mining techniques. Human-centric Computing and Information Sciences 8 (1), 3, https://doi.org/10.1186/s13673-_018-_0125-_x.

[16]. Moses, A., & Sarah, M. (2019) Analysis of Android Malware Detection Techniques: A systematic review. International Journal of cyber security and digital forensics. https://www.researchget.net/publication/320582721

Baffa Sani Mahmoud, et. al. "A Machine Learning Model for Malware Detection Using Recursive Feature Elimination (RFE) For Feature Selection and Ensemble Technique." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 24(1), 2022, pp. 23-30.