National College of Ireland

Multi Classifier Models using Machine Learning Techniques for Malware Detection

MSc Research Project
Cybersecurity

Janius Christabel Joseph
Student ID: X20112408

School of Computing
National College of Ireland

Supervisor: Dr. Vanessa Ayala-Rivera

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Janius Christabel Joseph
Student ID:	X20112408
Programme:	Cyber Security **Year:** 2021
Module:	MSc Research Project
Supervisor:	Vanessa Ayala-Rivera
Submission Due Date:	16/12/2021
Project Title:	Multi Classifier models using machine learning algorithms techniques for Malware Detection
Word Count:	5651 **Page Count: 20**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Janius Christabel Joseph

Date: 16/12/2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Multi Classifier models using machine learning algorithms techniques for Malware Detection

Janius Christabel Joseph
X20112408

Abstract

One of the most serious problems facing almost all sectors and industries is the battle against malware as new variants are being developed every day.

Malicious software is a serious threat to organizations, both public and private, as well as individuals at all levels. Because malware characteristics are constantly evolving, most existing solutions or anti-malware detectors are ineffective at identifying new strains of malware.

Most cutting-edge research today uses machine learning for malware detection. But the drawback with these methods is that they are mostly focused on binary classification and do not identify the kind of malware which has infected the systems. The approach used in this research aims to use a multi classifier to detect and classify malware.

Malware classification is approached using two techniques of binary and multi-class problems. The binary classification includes the differentiation between malicious and benign classes whereas the multi-classification includes classifying the malicious malware into Virus, Trojan, Spyware, Worms, Ransomware, and Adware type. Supervised learning approach and machine learning models like Random Forest model, Decision tree model, Support vector machine model, Naïve Bayes model, and K-Nearest Neighbour model is used for the classification of malware. The results show that Random Forest performs well in terms of Binary classification and the multi-classification problem with an accuracy of 95% and 91% respectively.

1 Introduction

Malware is a program that has been purposefully developed to engage in a variety of malicious behaviors, ranging from data theft to cyber espionage (Babaagba and Adesanya, 2021). Malware or malicious software is used by cybercriminals to cause financial or reputational damage. They could be used to infiltrate systems and gain access to confidential information. Malware nowadays is becoming more common and harder to identify due to the use of Obfuscation techniques used by attackers. This research uses machine learning algorithms for the detection and classification of the most common malware. Machine learning has triggered major changes in many fields including cybersecurity. It helps to filter through large amounts of data and identify potential threat patterns. This report will try to focus on the machine learning methods which will detect malware functioning in any Windows machine.

Different machine learning technologies have been extremely successful in the detection of malware functioning. Machine learning is a data analytics process that monitors visualization and other advanced processes in the targeted computer to acquire and integrate large-scale knowledge into the system automatically.

It depends on several models which work based on algorithms mainly used for machine learning. Some of the well-known machine learning techniques used by data analysts and scientists are regression, clustering, an ensemble of methods, neural nets and deep learning, dimensionality reduction, and classification. The four types of algorithms used in machine learning are unsupervised, supervised, reinforcement, and semi-supervised.

My experiment uses Supervised machine learning models using algorithms such as Random Forest, Naïve Bayes, Decision Tree, Support Vector Machine, and KNN for detection and classification of malware on a Windows machine. They are used to answer the following questions: **How supervised machine learning algorithms can be used to detect and classify Malware on Windows machines?** And **Which algorithm is best suitable for detection and classification problems?**

2 Literature Review

Machine learning techniques lead to reduced feature search spaces where the extraction method of feature space directly impacts the performance and detection of machine learning classifiers. Multiple feature selection reduces the number of features as well as their analysis to improve the level of accuracy of malicious applications and viruses. Analysis of multiple malware variants becomes difficult after a point due to which previous researchers have identified the classification of malware executable files extraction of byte code. Machine learning focuses on optimization of the procedures through which mining of malicious relations can be achieved and therefore the connections through data can be found out while computing major damages in network security **(Khammas, 2018).** The ability of n-gram feature space is an imperative attribute that represents the features in place of frequency by using the Boolean attribute. Machine learning opcode features have been used in the present study where detection of malware is done by n-gram extraction features. They have been combined with multiple classifiers raw byte and binary files. The authors in the paper have proposed that binary files will go through stages of n-gram features extraction then the first stage followed by second stage feature selection and ultimately into the classification process into a series of classes. In between these stages, there will be a classified model and a snort-sub signature model. The findings of the study show that the availability of new malware during software installation can be detected through binary files and this analysis is further minimized by search space features. Not only the accuracy level but also cross-validation of data set against malicious information is achieved through this methodology and it measures the capability of the techniques for malware detection at the network level. Adaboost IM used 10-fold cross-validation during training data set and is seen to do well during the different feature selection treatment experiment. Snort sub-signatures are the best machine learning classifiers since they enhance the performance through network-based applications and hot-based layering within the minimum feature space selected for malware detection.

The researchers have put forth an antivirus engine that can be used in business-level security systems for scanning malware using machine learning techniques. Though the system is

larger than normal machines due to the CPU power, it is very effective in Malware detection. The technique to identify the virus in this research study operates at the firewall level of the company network. They have used IAT for the extraction of PE header files which are then stored in a repository. The accuracy of their proposed methods is at 98% when compared to other machine learning algorithms such as Decision tree, Naïve Bayes and Random Forest. Although this model cannot be used by individuals or small organizations, it could be employed by bigger firms that are willing to invest more in security solutions. The researchers concluded that the proposed model, which is based on advanced data mining and machine learning methodologies, can detect malware **(Singhal, 2021).**

The research written by Sayadi et al **(2018)** proposes that hardware performance counters have a more robust need for the elimination of applications that exist by micro-architectural events captured during the running time of machine existence. The research has identified 8 techniques of robust machine learning methods for malware detection, and they have been characterized according to their levels of performance accuracy robustness, and hardware overheads. It has been found that ensemble learning has more commonly used algorithms that enhance the performance of malware detection through machine learning. Among them are Adaboost and bootstrap aggregation which are used for classification and problems of regression. The technique of bagging is also used with low classifiers where their variance is high, and the bias is low for the prediction of base learning, and they are highly dependent on how the data has been trained and managed in the system. The findings of the research study showed that J48, SMO, and Multilayer Perceptron have high performance when they are used with 8 and 16 hardware counters, but general ml classifiers show decreased potentials with fewer performance counters. This research has also given implications about hardware implementation which has been seen to detect malware functions in a magnitude of hierarchy rather than the need for latency during running time. Hardware implementation is deployed more by Xilinx Vertex 7 FPGA than others. The reason for this is that search arrangements make it possible for malware detection of low-level programs by reading a shared memory bus through the central processing unit of the system. It is also seen that ml classifiers have a good range of design areas and latency which gives better hardware solutions for efficient management besides neural networks are known for their delivery of high accuracy in hardware implementation cost. The findings of the research conclude that hardware-based detectors show unique performance in accuracy and robustness for malware detection and those techniques of boosting the performance get improved by 17% of classification after lower levels of performance counters. Thus, future architectures can modify the performance of ML classifiers through better malicious detection software.

The research paper has highlighted malware detection purposes to address malicious internet protocol issues through techniques such as cyber threat intelligence, machine learning, dynamic malware analysis, and data forensics. Use of bigger internet protocol reputation is associated with zero-day attacks and this is done by applying the decision tree technique. The research tries to highlight the complicated issues forensic that are associated with CVR risk and they tried to compare the different techniques of machine learning to attain precision-recall and better f-measure during the existence of proper forensic systems **(Usman et al., 2021).** There are comparisons between decision tree techniques and machine learning techniques where the former performs better due to a method of comprehensive analysis and provides better predictions of the accuracy of about 93.5 percent than machine learning techniques. The main goal of malware reporting is for analyzing those samples which cannot be retrieved due to memory corruption and as a result, there is a loss of information that cannot be discovered later. The analysis tool of real-time reporting has a simple stick

architecture method that tries to improve zero-day attacks by manually stating the dynamic analysis within a single component of real data. It forecasts the professional variants that are launched by malware towards the behavior of applications. The findings of the paper provide malware, dynamic, static, and family classification analysis. These parameters of findings try to assess the risk score of frequent malicious behavior and observed behavior through the various techniques of machine learning. An increasing number of features have been found to improve the performance of decision tree techniques as compared to other techniques of machine learning. Decision trees and machine-based services have an optimum approach in the examination of the purpose of the performance of networking data. The techniques that the research had thoroughly investigated and drew an efficient comparison are Support Vector Machine, Mini Batch K Means, Decision Tree, and Naive Bayes. Kill chain methodology should be implied to protect both internal and external data after their categorization through machine learning malware detectors.

The research paper that proposed that processes based on machine learning that are employed for the detection of malware have an analysis methodology where mal ID is a common segment for proper detection of malware files. The authors of the research paper have identified two Mal ID extensions that will provide ten times better performance measures and a high-rated decision tree methodology for classifying executable files. Reduced features of vector frameworks have many benefits among which one is feature selection and it is done for improvement of learning models performance and computation numbers through better learning speed enhancement of generalization capability and modified interpretation of complex models **(Tahan et al., 2012)**. The mixtures of features that are found in processes of malware detection are mostly API execution representation through strings and program strings. The detection process starts with the extraction of API functions, and they look for consecutive bytes that are seen as a string. Application of higher platforms for level development is a common feature in all parts of malware similar to all kinds of system software. The peeper proceeds from setup phase-detection face to algorithms used by MI ID basic detectors. The combination of gain ratio feature selection with decision tree provided the best performance under all conditions of data set content set training to resize data set manipulation. The rotation forest boosting method was seen to perform better than Adaboost MI and non-boosting machine learning methods such as J48. The comparison of performance between preliminary and complex methods for detection is seen as average entropy. A decrease in posterior cross-entropy is the result of the probability that the output of all experimental testing conditions has been higher in small training sizes of a set. Mal ID + RF are seen to outperform all the other methods and is eventually the highest efficient detector in all the conditions. The basic model of Mal-ID has at least AUC functioning contrary to others sins despite being a basic model it is identified as a discrete classifier for highest and complex malicious behavior.

Increased use of electronic devices such as mobiles has made it easier for hackers to break the network security of the system and lead to more cyber theft, credential theft, malicious advertising, and surveillance. It is the responsibility of machine learning methodologies to detect such attacks and to prevent further possibilities by using classifiers. Android malware detection techniques are a bit different than computer detection malware and the present study, therefore, tries to highlight the strengths and weaknesses that are involved in the detection of such electronic device malicious behavior **(Senanayake et al., 2021)**. The built-in security in the Android system is in what sandboxing technique and permission system but there are risks and bugs implied in the application. Presence of Linux environment user's unique identifiers and without granting permission it becomes difficult for reconfiguration of

applications as well as access of system resources. Malware detection functions in Android mainly try to address the weaknesses and security flaws that can become a threat to the user. They are mostly social engineering attacks, network attacks, third-party library vulnerabilities, virtualization vulnerabilities, Android debug bridges, and kernel vulnerabilities. Malware attacks are mainly for peace of malicious codes and even intent on the user and they are done through illegal and unethical activities violating integrity confidentiality and availability of information to third party users. Search malware activities are much easier to perform than in computer systems because the browser's network and devices are less privileged and have reduced acquisition modes. Thus, to address search issues, machine learning which is artificial intelligence is used for explicitly performing data mining and data analysis for the determination of malicious content from the original one. The authors in the present study tried to investigate the reviews that had been collected in machine learning methods and how they can be used for analyzing APK Android versions to detect vulnerabilities and malicious application behavior. The findings from the current systematic review paper show that Android malware technologies are highly evolving and there are necessary emerging processes to address such detection methods. They are more accurate than traditional techniques of machine learning and are also beneficial for comprehensive reviews of the contribution of artificial intelligence in this particular area.

Motivation to use Android malware detection is now widely emerging because of large-scale inventions of Android applications and the probability of misuse to give a threat to the users. The study by Memon et al., **(2019)** found out that the mechanism of gradient boosted tree has the highest precision as well as accuracy in detection of Android malware applications. The study had 83 attributes out of which 29 were identified as suitable features for malware detection. This indicates a positive stance for extensive evaluation in real cloud-based application systems and malware detection by various technologies along with their harmful effects can also be contained through emerging machine learning techniques. Android malware types are mainly hybrid and dynamic where the former detection methods require taint analysis, emulation-based detection, and anomaly-based detection while the latter only requires a machine learning approach. Hybrid animals have a high false-positive rate and they are very slow to detect because of high training costs due to the use of techniques. The most commonly used machine learning techniques for Android malware detection are Bayes Net, Naive Bayes, Support Vector Machine, Logistic Regression, Decision tree, and Adaboost. The current study shows that the support vector machine has the lowest accuracy and highest false-positive rate due to skewed or imbalanced datasheets. Random forest decision trees and gradient boosted trees word the highest classifiers for detecting label malicious label benign and false-positive rate. This indicated that they can create services even on a private cloud where the application input of feature vectors can be tracked from Android smartphones and it will enable to classify either the malicious content as benign or malicious highly to an extent. Machine learning classifiers, therefore, used this characterization to give information to the users on private clouds such as Apache Spark which had been used in the present study about how to avoid the flow of memory and storage falsification in their device.

The authors in the present study **(Taheri et al., 2020)** identified detection methods of malware behaviour with the use of hamming distance which is all nearest neighbours, first nearest neighbours, k-medoid based nearest neighbours, and weighted all nearest neighbours. The findings of the study showed that the effectiveness of the classifiers carried out higher performance than classification and detection algorithms under the name of mixed and separated solutions, program dissimilarity measure based on entropy and fulleroid algorithms. Permission features and API intent confirm that the accuracy rate of the algorithm which was

proposed by the study is 90% higher than the existing state of art solutions provided by previous studies. The authors also suggested a new system of detection for malware known as Anastasia for identification of malicious samples through API features system command and intent of the sample. Android cloud applications required delicate feature types through which signature approaches can try to tactfully detect malware and one such automatic approach of signature generation is Andrew similar where it uses static synthetic features derived from applications default in Android smartphones. These approaches are mainly for going out of the box and comparing state-of-art methods that have already been applied for malware detection through a combination of classification and clustering learning methods. It can be evaluated from the findings of the study that hamming distance has been able to identify similar samples and present methodologies for Android application malware with high precision and rates of recall. Not only do they validate the algorithms, but they also approach KNN based solutions to demonstrate the appropriate percentage of malware detection cases.

Thus, it can be concluded from the large body of research studies conducted on malware detection that the use of machine learning strategies is an imperative approach where it will recognize the areas of trivial as well as major issues that any system is facing due to cyber exploitation. Malware detection becomes very difficult also because of the presence of advanced technologies and emerging ideas by hackers who try to corrupt the system with difficult coded virus worms and Trojan horses. Therefore, to address search major cyber threat issues machine learning technologies also need to advance their approaches to delivering the highest accuracy rate and robustness in the process of malware detection.

3 Research Methodology

Malware detection systems can be used as an effective tool to protect infrastructure against different types of malware-based attacks. The below workflow diagram helps to understand the steps which were taken to implement this solution.

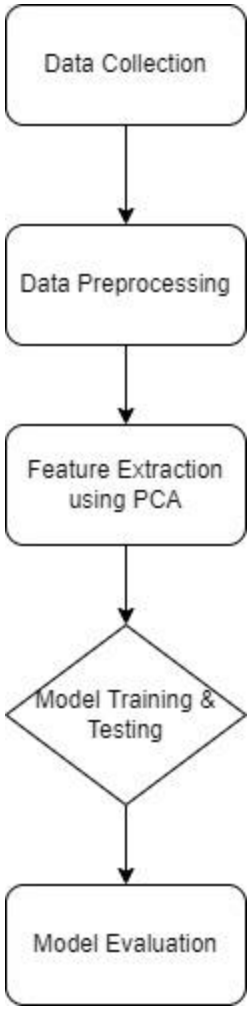

Figure 1: Methodology Steps

Data Set Collection

The dataset for this experiment was obtained from Kaggle, an open-source platform. The dataset chosen contains 19612 rows and 79 columns of malware and benign samples (Benign & Malicious PE Files, 2022).
The dataset is split into 80% for training and 20% for testing purposes.

Data Preprocessing

This is an important step to ensure the dataset provides accurate results. The initial step would be to use Python's built-in methods to check for null and missing values in the rows (this dataset has no missing values). The data is also analyzed to make sure all the data types are integers except for one column.
Statistical data analysis is carried out to identify outliers in the data and prevent model biases. Correlation is performed to check if variables are dependent on each other or not.

Data Feature Visualization

Using some of the prominent features from the correlation after label encoding the energy density plots are marked.

Feature Extraction

PCA in character selection: Principal Component Analysis (PCA), was used for feature reduction as the dataset had many un-correlated data features. It is a technique improves interpretability while minimizing risk of data loss.
PCA will reduce the 79 features into components based on the most contributing features. The models are then built based on these components to arrive at the results.

Model Training

The below algorithms are used for the training and testing of the dataset.

1. Random forest method
2. Decision tree method
3. Support Vector machine method
4. Naïve Bayes method
5. K-Nearest Neighbour method

Binary Problem – Using the above-mentioned algorithms, the binary classification problem is addressed by identifying the malware from benign samples.

Multi-class Problem – At this stage, labels are assigned to the malware samples as Adware, Virus, Spyware, Trojan, Worm, and Ransomware. Once again, the dataset is split into training and testing for model building.

Model Evaluation

The models are evaluated using the confusion matrix and the below metrics

Accuracy = TP + TN / TP + FP + TN + FN

Precision = TP / TP + FP

Recall = TP / TP + FN

F1 score = 2* (Precision*Recall / Precision + Recall)

4 Design Specification

Five machine learning algorithms are used for the detection of files as malicious and benign and classification into their respective malware family.

Feature selection is done using PCA (Principal Component Analysis) as the dataset has many un-correlated data attributes.
The model with the maximum accuracy and best performance is selected to be the final model.

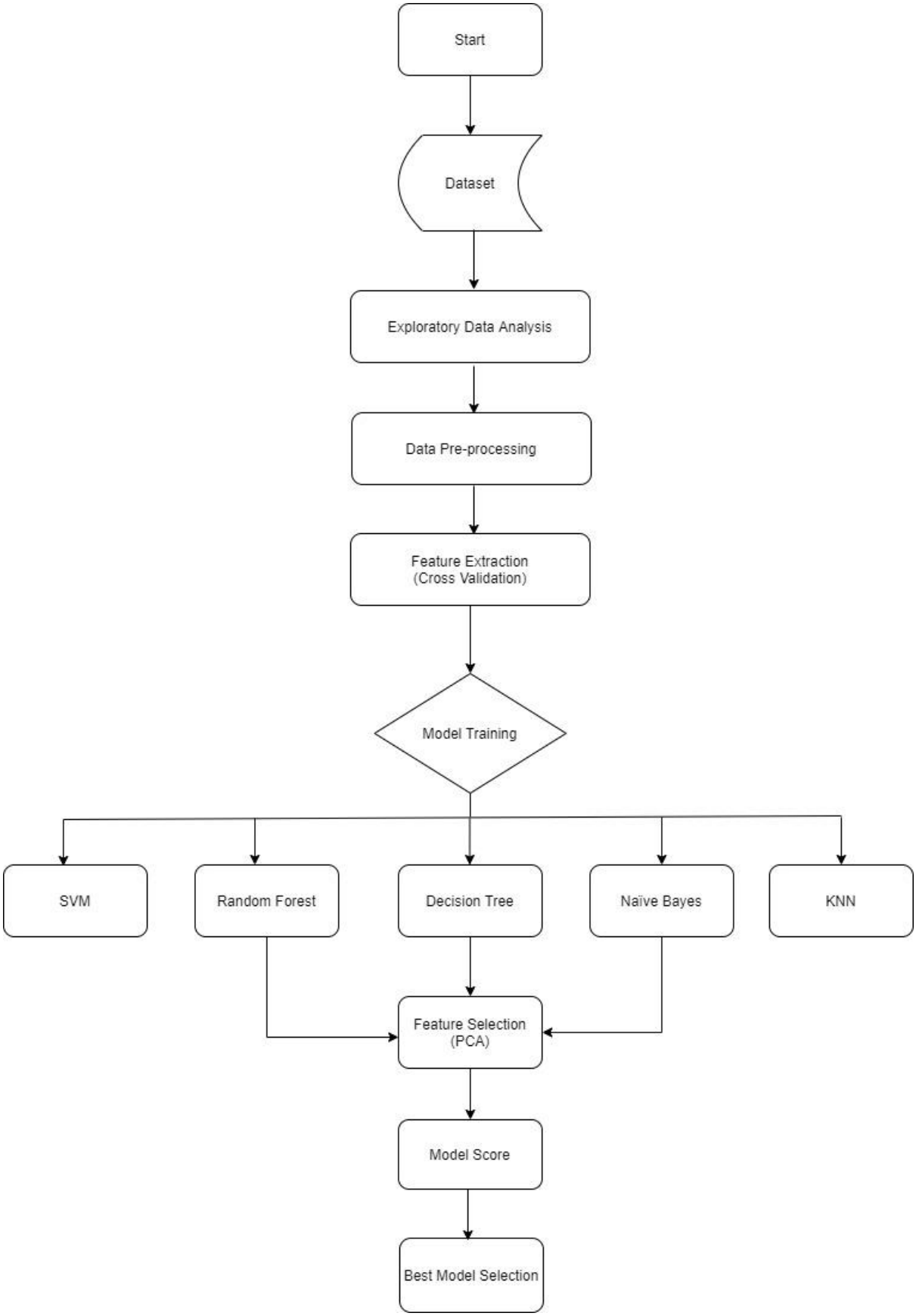

Figure 2: Workflow diagram

4.1 Random forest algorithm:

The algorithm does not work with data sets containing categorical data, so it requires pre-processing databases, such as converting ordinary data into numbers. After converting all the categorical strings of the letter in the database to numbers, the user can use that database with the algorithm.

4.2 Decision tree algorithm:

The Decision tree classifier is basically used for the classification because of its ability to avoid model overfitting with the selected dataset. At each stage or node of the decision tree used for classification, try to create a condition in the elements to separate all the labels or classes in the database in order to be completely clean.

One of the biggest challenges of the decision tree is that it leads to data filling. The main reason for this is to develop an environment-based approach to training data. Therefore, it is firmly established in the training data. Now, when conditions are mostly used in train data on a tree when it is deeper, it equals data. But point by point, it begins to take into account very small changes in some aspects that provide less data gain, especially points that weaken the normal performance of the model and lead to outliers in the experimental data. We may limit these terms by limiting the availability of information, so if the information provided by the condition is less than the value provided, we do not consider the condition. This prevents a little overfitting and helps in creating a standard design. The Decision tree classifier is basically used for the classification because of its ability to avoid model overfitting with the selected dataset.

4.3 Support Vector Machine algorithm:

SVM are supervised Machine learning models which are primarily used for regression and classification problems. These models are also useful when the number of dimensions is more than the number of samples.

4.4 Naïve Bayes algorithm:

The Naïve Bayes is a probabilistic supervised learning algorithm that solves classification problems based on the Bayes theorem.

This algorithm has been used effectively for a variety of applications, but it excels at natural language processing (NLP) problems.

4.5 K- Nearest Neighbour algorithm:

The K-NN algorithm is a simple supervised machine learning algorithm that can be used both for classification and regression. It's an instance-based algorithm. So instead of estimating a model, it stores all training examples in memory and makes predictions using a similarity measure.

KNN (K-Nearest Neighbor) classifier is utilized for its feature boosting technique. Significant features are desirable to obtain the best model performance.

All the above-mentioned models will be compared based on their evaluation metrics and overall model performance.

5 Implementation

Using the dataset collected and stored in a comma-separated file (CSV) format which is loaded into Google Colab, using Python code for data manipulation.
The next step in the workflow is to explore the data with the aid of exploratory data analysis steps and draw valuable insights like outliers in the data attributes, significant data feature identification, and selection of the target data attributes.

Primary data pre-processing activities, where the data is cleaned, NULL values are removed from the data columns, NaN, and missing values are replaced with the mean of the data column. Statistical analysis of the dataset is done to analyze the mean, median, 25% quartile, 50% quartile, 75% quartile of the individual data attribute. The statistical analysis helps analyze the numerical data attributes to observe their maximum and minimum value for machine learning model training, testing, and evaluation.

The problem will be analyzed using two aspects of classification, first, the PE files will be classified as malicious and benign (Binary classification) and second the malicious files are classified according to the malware type by which the file has been affected (multi-class classification).

Classification:
a) unweighted: output the most common classification among the k-nearest neighbors
b) weighted: sum up the weights of the k-nearest neighbors for each classification value, output classification with the highest weight

The random forest classifier along with the other four different classifiers is used for the classification of the malicious and benign labeled dataset, further classifying the malicious data according to their respective labels and malware type before converting all the numerical and categorical data attributes.

Then training of the above-mentioned data using machine learning algorithms is done. Using feature selection techniques to select the significant features from the data. The feature selection is necessary in this case because of the large size of the data. The dataset consists of 50+ data attributes that are to be considered for classification purposes. Using Principal Component Analysis (PCA) for the feature selection along with correlation matrix which depicts the nature of dependent or independent data attributes. In this case the PCA reduced the 79 features into 10 components with the features most contributing.

Feature selection is the process by which you automatically select or implement attributes that significantly contribute to the estimation of output variables. Having unimportant features in the data reduces the accuracy of the model and allows the model to learn based on unnecessary features.

The last step of the workflow involves the selection of the best-performing model based on evaluation metrics. They are not purely based on accuracy alone but others which can help determine the best performing model.
The evaluation metrics that are considered for the comparative analysis for these models are accuracy score (testing), precision value, recall value, F1-Score, confusion matrix, and the classification report.

The models for binary and multi-classifiers are chosen based on the best performance and the previously indicated evaluation metrics. The aim of classifying malware files into their type is achieved using the machine learning approach.

6 Evaluation and Results

Using Accuracy, Precision, Recall and F1 score the performance of the Binary and Multi classifier models are evaluated.

6.1 Random Forest Algorithm Models

Binary Classification model

	precision	recall	f1-score	support
Benign	0.91	0.91	0.91	1004
Malware	0.97	0.97	0.97	2919
accuracy			0.95	3923
macro avg	0.94	0.94	0.94	3923
weighted avg	0.95	0.95	0.95	3923

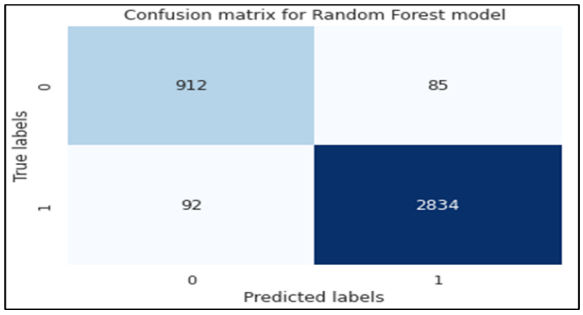

Figure 3: Classification report for random forest with binary classes and Confusion Matrix

Multi-classification model

	precision	recall	f1-score	support
Adware	0.99	0.91	0.95	90
Virus	0.93	0.95	0.94	3044
Spyware	0.80	0.73	0.77	786
Trojan	0.00	0.00	0.00	1
Worm	1.00	1.00	1.00	1
Ransomware	0.00	0.00	0.00	1
accuracy			0.91	3923
macro avg	0.62	0.60	0.61	3923
weighted avg	0.90	0.91	0.91	3923

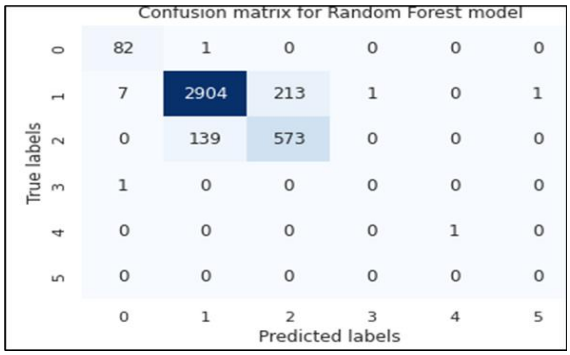

Figure 4: Random Forest model fitting for Multi classification & Confusion Matrix

6.2 Decision Tree Algorithm Models

Binary Classification model

	precision	recall	f1-score	support
Benign	0.89	0.91	0.90	1004
Malware	0.97	0.96	0.97	2919
accuracy			0.95	3923
macro avg	0.93	0.93	0.93	3923
weighted avg	0.95	0.95	0.95	3923

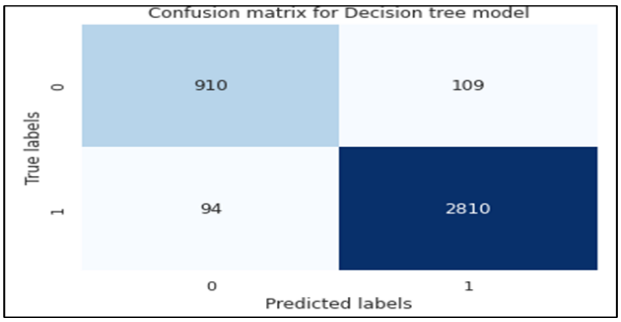

Figure 5: Classification report for Decision Tree with binary classes and Confusion Matrix

Multi-classification model

	precision	recall	f1-score	support
Adware	0.96	0.90	0.93	90
Virus	0.93	0.94	0.93	3044
Spyware	0.75	0.73	0.74	786
Trojan	0.00	0.00	0.00	1
Worm	1.00	1.00	1.00	1
Ransomware	0.00	0.00	0.00	1
accuracy			0.89	3923
macro avg	0.61	0.59	0.60	3923
weighted avg	0.89	0.89	0.89	3923

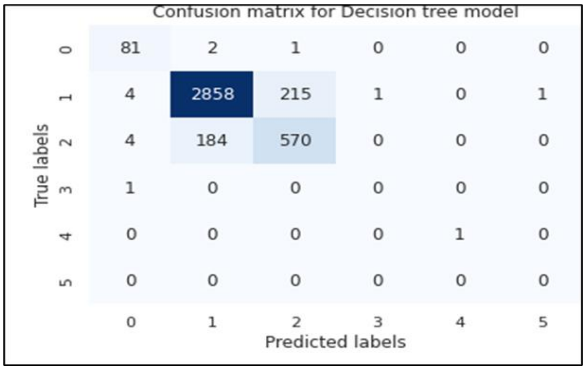

Figure 6: Decision Tree model fitting for Multiple classifications & Confusion Matrix

6.3 Support Vector Machine Model

Binary Classification model

	precision	recall	f1-score	support
Benign	0.93	0.01	0.03	1004
Malware	0.75	1.00	0.85	2919
accuracy			0.75	3923
macro avg	0.84	0.51	0.44	3923
weighted avg	0.79	0.75	0.64	3923

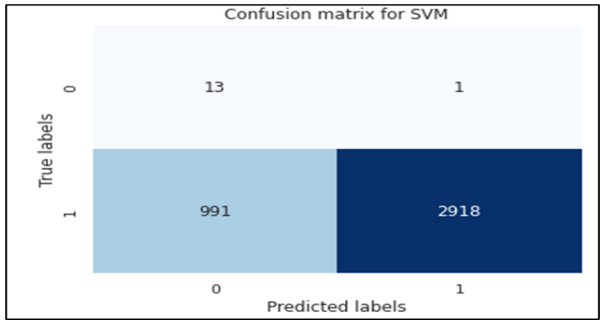

Figure 7: Classification report for Support Vector Machine with binary classes and Confusion Matrix

Multi-classification model

	precision	recall	f1-score	support
Adware	1.00	0.09	0.16	90
Virus	0.78	1.00	0.88	3044
Spyware	1.00	0.03	0.06	786
Trojan	0.00	0.00	0.00	1
Worm	0.00	0.00	0.00	1
Ransomware	0.00	0.00	0.00	1
accuracy			0.78	3923
macro avg	0.46	0.19	0.18	3923
weighted avg	0.83	0.78	0.70	3923

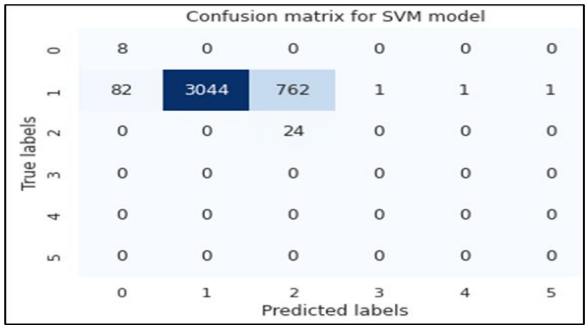

Figure 8: Support Vector Machine model fitting for Multiple classifications & Confusion Matrix

6.4. Naïve Bayes Model

Binary Classification model

	precision	recall	f1-score	support
Benign	0.27	1.00	0.43	1004
Malware	0.99	0.09	0.16	2919
accuracy			0.32	3923
macro avg	0.63	0.54	0.29	3923
weighted avg	0.81	0.32	0.23	3923

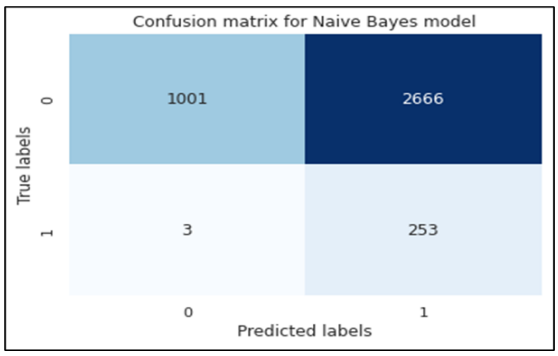

Figure 9: Classification report for Support Vector Machine with binary classes and Confusion Matrix

Multi-classification model

	precision	recall	f1-score	support
Adware	0.00	0.00	0.00	90
Virus	1.00	0.07	0.13	3044
Spyware	0.48	0.73	0.58	786
Trojan	0.00	0.00	0.00	1
Worm	0.00	1.00	0.00	1
Ransomware	0.00	1.00	0.00	1
accuracy			0.20	3923
macro avg	0.25	0.47	0.12	3923
weighted avg	0.87	0.20	0.22	3923

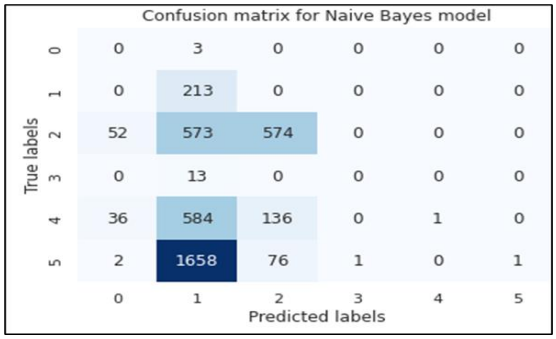

Figure 10: Naïve Bayes model fitting for Multiple classification & Confusion Matrix

6.5. K-Nearest Neighbor Model

Binary Classification model

	precision	recall	f1-score	support
Benign	0.90	0.90	0.90	1004
Malware	0.97	0.96	0.97	2919
accuracy			0.95	3923
macro avg	0.93	0.93	0.93	3923
weighted avg	0.95	0.95	0.95	3923

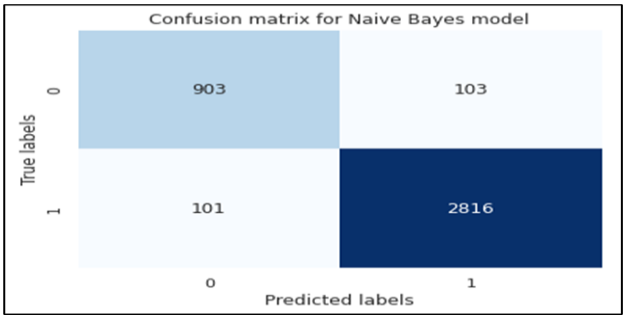

Figure 11: Classification report for Binary Classification with binary classes and Confusion Matrix

Multi-classification model

	precision	recall	f1-score	support
Adware	0.97	0.92	0.94	90
Virus	0.93	0.94	0.93	3044
Spyware	0.75	0.73	0.74	786
Trojan	0.00	0.00	0.00	1
Worm	1.00	1.00	1.00	1
Ransomware	1.00	1.00	1.00	1
accuracy			0.89	3923
macro avg	0.77	0.76	0.77	3923
weighted avg	0.89	0.89	0.89	3923

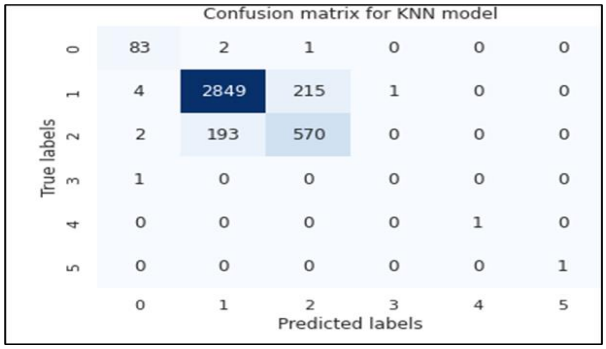

Figure 12: Classification report for K-Nearest Neighbor with binary classes and Confusion Matrix

6.1 Discussion

Figure 13 depicts the evaluation metrics obtained for all the five machine learning models for binary classification.

Models	Accuracy score	Precision value	Recall value	F1-Score	Support value
Random Forest Algorithm	0.95	0.97	0.97	0.97	2919
Decision tree algorithm	0.75	0.75	1.00	0.85	2919
Support vector	0.75	0.75	0.01	0.02	2919

machine algorithm					
Naïve Bayes algorithm	0.32	0.99	0.09	0.16	2919
K-Nearest Neighbor algorithm	0.95	0.97	0.96	0.97	2919

Figure 13: Evaluation Metrics of Binary classifier models

From Figure 13, Random Forest and KNN algorithms perform well in terms of binary classification.

Figure 14 depicts the evaluation metrics obtained for all the five machine learning models for multiple classifications.

Models	Accuracy score	Precision value	Recall value	F1-Score	Support value
Random Forest Algorithm	0.91	0.93	0.95	0.94	3044
Decision tree algorithm	0.89	0.93	0.94	0.93	3044
Support vector machine algorithm	0.78	0.78	1.00	0.88	3044
Naïve Bayes algorithm	0.20	1.00	0.07	0.13	3044
K-Nearest Neighbor algorithm	0.89	0.93	0.94	0.93	3045

Figure 14: Evaluation Metrics of Multi classifier models

From Figure 14, Even for the multi-classification problem, the Random Forest algorithm performs better.

7 Conclusion and Future Work

This research paper proposes a novel approach where it can perform detection and classification of malware at high accuracy. It uses five algorithms such as Random Forest, Decision Tree, Support Vector, Naïve Bayes, and KNN for the models.

Future work involves implementing a deep neural network (CNN) for the classification of more complex malware files. Also, getting more data from other sources is required so that model could be trained and tested on all the available malware types. This system could also be converted to an API (Application Programming Interface) which can be used and integrated with other languages and platforms as well.

References

[1] Taheri, R., Ghahramani, M., Javidan, R., Shojafar, M., Pooranian, Z. and Conti, M., 2020. Similarity-based Android malware detection using Hamming distance of static binary features. *Future Generation Computer Systems*, *105*, pp.230-247.

[2] Memon, L.U., Bawany, N.Z. and Shamsi, J.A., 2019. A comparison of machine learning techniques for android malware detection using apache spark. *Journal of Engineering Science and Technology*, *14*(3), pp.1572-1586.

[3] Senanayake, J., Kalutarage, H. and Al-Kadri, M.O., 2021. Android Mobile Malware Detection Using Machine Learning: A Systematic Review. *Electronics*, *10*(13), p.1606.

[4] Khammas, B., 2018. Malware detection using sub-signatures and machine learning techniques. *Journal of Information Security Research*, *9*(3), pp.96-106.

[5] Tahan, G., Rokach, L. and Shahar, Y., 2012. Mal-id: Automatic malware detection using common segment analysis and meta-features. *Journal of Machine Learning Research*, *13*(4).

[6] Gupta, D. and Rani, R., 2020. Improving malware detection using big data and ensemble learning. *Computers & Electrical Engineering*, *86*, p.106729.

[7] Usman, N., Usman, S., Khan, F., Jan, M.A., Sajid, A., Alazab, M. and Watters, P., 2021. Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics. *Future Generation Computer Systems*, *118*, pp.124-141.

[8] Sayadi, H., Patel, N., PD, S.M., Sasan, A., Rafatirad, S. and Homayoun, H., 2018, June. Ensemble learning for effective run-time hardware-based malware detection: A comprehensive analysis and classification. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)* (pp. 1-6). IEEE.

[9] Liu, X., Lin, Y., Li, H. and Zhang, J., 2020. A novel method for malware detection on ML-based visualization technique. *Computers & Security*, *89*, p.101682.

[10] Sayadi, H., Houmansadr, A., Rafatirad, S. and Homayoun, H., 2018, May. Comprehensive assessment of run-time hardware-supported malware detection using general and ensemble learning. In *Proceedings of the 15th ACM International Conference on Computing Frontiers* (pp. 212-215).

[11] Babaagba, K. and Adesanya, S., 2021. A Study on the Effect of Feature Selection on Malware Analysis using Machine Learning. [online] Available at: https://www.napier.ac.uk/~/media/worktribe/output-2793726/a-study-on-the-effect-of-feature-selection-on-malware-analysis-using-machine-learning.pdf

[12] Singhal, P., 2021. *Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks*.

[13] Kaggle.com. 2022. *Benign & Malicious PE Files*. [online] Available at: <https://www.kaggle.com/amauricio/pe-files-malwares?select=dataset_malwares.csv> [Accessed 7 January 2022].