Malware Classification Using Machine Learning Approach

Submitted By

Md. Mahfuj Hasan Shohug

ID: 181-35-2376

Batch: 25th

Department of Software Engineering

Daffodil International University

Supervised By

Ms. Farzana Sadia

Assistant Professor

Department of Software Engineering

Daffodil International University

This thesis report was submitted in order to meet the requirements for a Bachelor of Science in Software Engineering degree.

APPROVAL

This thesis titled on "Malware Classification Using Machine Learning Approach.", submitted by Md. Mahfuj Hasan Shohug, Id- 181-35-2376 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of Bachelor of Science in Software Engineering and approval.

Dr. Imran Mahmud — **Chairman**

Associate Professor and Head

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Afsana Begum
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Tapushe Rabaya Toma
Senior Lecturer
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Prof. Dr. Md. Saiful Islam — **External Examiner**

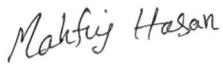
Professor

Institute of Information and Communication Technology (IICT)

Bangladesh University of Engineering and Technology (BUET)

DECLARATION

I hereby declare that this thesis (Malware Classification Using Machine Learning Approach) has been done by me under the supervision of Ms. Farzana Sadia. Faculty of Science and Information Technology, Department of Software Engineering (SWE), Daffodil International University (DIU). It is additionally declared that neither this thesis nor any component has been submitted elsewhere for the award of any degree. All declarations are fully verified for completeness and the validity of their data element contents.



Md. Mahfuj Hasan Shohug

ID: 181-35-2376

Batch 25th

Department of Software Engineering

Faculty of Science & Information Technology Daffodil International University

Certified by:

Ms. Farzana Sadia

Assistant Professor

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

ACKNOWLEDGEMENT

At the outset, I express my gratitude to Almighty Allah for granting me the capacity to complete the final thesis, and I would like to express my gratitude to my family, who have always been supportive and believed in me.

Finally, I'd like to express my gratitude to my supervisor, Ms. Farzana Sadia, Assistant Professor Faculty of Science and Information Technology, Department of Software Engineering, Daffodil International University, for allowing me to work on this project and for providing me with valuable guidance and advice on issues that arose during the implementation of this thesis.

Table of Contents

ABSTRACT

Malware classification is essential for tracing the source of computer security threats. On the Internet, malware evolves at a rapid rate, and the bulk of undiscovered malware is developed from known malware. The number of malwares has expanded considerably in recent years, posing a serious security threat to financial institutions, businesses, and individuals. To stop malware from spreading, new methods for quickly recognizing and classifying malware samples so that their behaviour can be investigated are needed. In current Internet age, many virus attacks occur, posing serious security risks to financial institutions and everyday customers. The total number of malware occurrences has undoubtedly increased considerably over time. Here I use five machine learning classification model for the fast time in this dataset. I am classified according to the 54 correlated features with data visualizing, resizing and prepressing and finally proposed the best model for detection malware and model preparation method into many parts in this work. With almost 99% accuracy, the Random Forest Classifier outperforms. Second, with a score of 97 percent, K-Neighbors Classifier comes in second place in terms of malware classification accuracy. The rest of the models are less accurate.

Keywords: Malware Classification, K-Neighbors, Random Forest.

CHAPTER 1

INTRODUCTION

1.1 Background

Malware is malicious software (such as viruses, worms, trojan horses, and spyware) that causes damage to computer systems or conducts harmful actions. Many virus attacks occur in this Internet age, posing severe security dangers to financial institutions and common consumers. It is undeniable that the total number of malware incidents has risen dramatically over time. According to Symantec, more than 357 million new malware types were discovered in 2016.The significant use of obfuscation techniques by malware producers is one of the key causes for the enormous volume of malware samples. This means that dangerous files from the same malware family (i.e. identical code and common origin) are continually updated and/or disguised.

Malware classification is crucial for determining the origins of computer security attacks. Static analysis methods are fast in classification, but they are ineffective in some malware that uses packing and obfuscation techniques; [4] dynamic analysis methods have superior universality for packing and obfuscation, but they will increase classification costs. The introduction of numerous automated technologies has demonstrated that malware evolves on the Internet at a far faster rate than most people believe.

In 2017, Kaspersky Labs detected 15,714,700 hostile objects [28], whereas McAfee Labs detected 79 million malicious objects per day in 2018 Q1 (Q1 indicates first quartal), up from 45 million in 2017 Q4 (Q4 means fourth quartal) [29]. Despite the fact that malware evolves at an increasing rate on the Internet, the majority of unknown malware is derived from known malware. Although static features can be used to classify most malware, the prevalence of packing and obfuscation techniques makes it easy to create malware with consistent behavior but inconsistent static features. Such malware necessitates dynamic analysis. Although dynamic analysis is more effective than static analysis in behavioral analysis, it is also more expensive [30]. As a result, finding an effective combination technique to overcome these challenges is required. Previous malware classification research reveals that malware samples often belong to a family with similar behaviors, implying that most new malware is really a variation of old malware [31]. As a result, the idea of developing a system that can efficiently classify malware based on its family, regardless of whether it is a variant, appears particularly promising and a means of dealing with malware's rapid proliferation.

In recent years, the volume of malware has increased dramatically, posing a severe security danger to financial institutions, enterprises, and individuals. New ways for promptly

identifying and classifying malware samples so that their behaviour can be examined are required to stop the spread of malware.

In this paper we will deploy different machine learning model to detect malware. We will apply different machine learning model to detect malware and find which model performs base.

1.2 Motivation

Recently there are lots of cyberattack incidents happened al over the world. Also, it happened in Bangladesh. The massive SolarWinds cyberespionage attack, which was discovered in December 2020, [29] breached U.S. federal agencies, infrastructure, and private corporations in what is thought to be one of the worst cyberespionage attacks ever perpetrated against the United States. SolarWinds, an Austin-based IT management software company, was hit by a supply chain attack on Dec. 13, 2020, compromising updates for its Orion software platform. Threat actors inserted their own malware, now known as Sunburst or Solorigate, into the updates that were distributed to many SolarWinds customers as part of this attack. The cybersecurity firm FireEye was the first confirmed victim of this backdoor, revealing on Dec. 8 that it had been hacked by suspected nation-state hackers. Also, in Bangladesh our central Bank was attracted by the foreign hacker and they stole a lots amount of money from Bangladesh bank. Those incidents give me the motivation to make a machine learning model for detecting most dangerous malware and give a safe solution for that malicious software.

1.3 Problem Statement

In any kind of software there have lots of feature but the most useable and most relatable 54 features I am use in my own thesis. Here I am classified the legitimate (0 for Malware and 1 for Safe Software) according to those using features. And after that I build 4 machine learning model and detect which software is malware and which is not.

1.4 Research Question

The research question for my won thesis was given bellow:

- ✓ RQ1: How machine learning model detects perfectly from those given multiple features?
- ✓ RQ2: How to classify malware software in two categories (Legitimate or Not)?

1.5 Research Objective

The main goal of this thesis is to build an automatic malware detection system from using my collected train datasets with 54 different features. The following are the goals of my thesis:

- Divide all rows from my collected dataset into two categories.
- Increase training dataset.
- Train machine learning classification models in this dataset.
- Save models' progress.
- Focusing for the classification for that malicious software which is affecting or not.
- For detecting those malwares using the multiple Machine learning models.
- After those ML classification models classification models completing define then best model for detecting those malwares.
- Save best weight for predicted classification.
- Predict malware software from unknown test dataset.

1.6 Research Scope

The scope of this thesis is exiting helpful for:

- This system helps to safe any kind of important software.
- If any kind of hacker try to installed a malware on any kind of device, then notify before the detection.
- Focusing for the classification for that malicious software which is affecting or not.
- For detecting those malwares using the multiple Machine learning models.
- After those ML classification models and deep learning classification models completing define then best model for detecting those malwares.

1.7 Challenges

There are lots of challenges for complete my thesis perfectly. As per my research method the most challenge from my perspective it was collecting this huge malware classification dataset. Here have some most flowing challenges:

1.1.1 Data Collection

This data came from the "Kaggle" website competition dataset "Malware Detection- Make your own Malware security system, in cooperation with malware security partner Max Secure Software," which I discovered. There are almost 2 lakh sixteen thousand data rows with 54

features in this dataset. This is a large dataset, and I did not find this dataset connected to malware classification work in my literature review.

1.1.2 Level Generation

Here in this large dataset, it was difficult to specified the dataset category for which feature value define that which is malware or which is not.

1.8 Contribution

Here in my thesis, I am contributing for not only to detect the malware but also using this large data in different classification models and chose the best model which is predict most accurately. To improve accuracy and model performance, this paper used a variety of strategies. My study's main goal is to improve the model so that it can also find the best accurate estimate result from the training dataset.

1.9 Thesis Organization

In the following chapter, I looked at other studies on the same topic that had research gaps. My suggested research method was presented in the final chapter. The results of the analysis are discussed in the fourth chapter. Finally, I go over the observations and recommendations from the fifth chapter, which include assumptions, limitations, and future research.

CHAPTER 2

RELATED WORK

2.1 Introduction

A literature review is a survey of scholarly articles, books, dissertations, conference proceedings, and/or other materials that have been published. The review summarizes, describes, and evaluates a topic, issue, or research area. It's not to be confused with a work review, which is a less formal format for summarizing a piece of work.

2.2 Paper Review

Di Xue et al (2019). author propose a classification method based on probability scoring and machine learning, setting probability thresholds to connect static and dynamic analysis, speeding up the static analysis process, and using dynamic analysis appears to have greater adaptability to fuzzy processing. They define malware classification methods that primarily focus on feature engineering, which requires extracting features from malware or visualization images, such as API calls, system calls, and a variety of other features. [1]

Kalash et al (2018). In this paper, authors proposed a deep learning model which is Convolutional Neural Networks (CNN) model and try to classify the malware through the two types of image datasets. In their CNN model they found the highest accuracy for the malware detection. They claim that using Malimg and Microsoft malware datasets, their method outperforms the current state-of-the-art performance.[2]

Milosevic et al (2017). In their research paper, they present two machine learning aided approaches for static analysis of the mobile applications, one is based on Permission-based analysis and second is Source code-based analysis. Their source code-based classification had an F-measure of 95.1 percent, whereas the approach that used permission names had an F-measure of only 89 percent. They also claim that for improve the all result for those two aided approaches can be increase while those data sets size will be bigger labelled balanced data set.[3]

Ucci et al (2019). On the paper, they collected data from 3 country to make method for automatically detection of road damage. The data was collected from India, Japan, Czech Republic. They arrange a competition to choose the best result. This was GRDDC (global road detection) and competition was organized by IEEE. Data was divided into 3 part. Train, test1, test2. From the competition the best result which was taken was done by YOLO ensemble learning and get 0.67 F1 score on test1 and 0.66 F1 score on test2.[4]

Fu et al (2018), In this research paper, they visualize malware as RGB-coloured images and extract global features from the images and use three classification machine models with merge the global features and local features to perform malware classification. For visualize their

dataset they use feature extraction method with global feature and texture feature and compared those three models. They also characterization of the global features of malware to extract texture features and colour features, respectively. They also provide static features and dynamic features for classification and also, they extracted local features from code sections features and local features obtained the highest accuracy.[5]

Gibert et al (2020), Here in this paper, they present a systematic review of malware detection and classification approaches using machine learning. They sum up, a total of 67 research papers for tackling the problem of malware detection and classification on the Windows platform are reviewed. There are four main contributions of their work. Most of the time they focus dynamic and static features and then using multiple models detect the malware.[6]

Zhang et al (2016), In this paper, mainly they use signature-based detection system. And also use to counter large volumes of malware variants, machine learning techniques have been applied for automated malware classification. The results of their experiment show that our method can effectively and efficiently classify malware samples into their respective families, even when the training set is skewed. They mainly come to the conclusion that XGBoost and Extra Tree Classifier offer promising results for big data applications.[7]

Kang et al (2016), Here in this paper they found to be 8.2% of the time, and no case of neomycin-resistant illness was found during the trial. The only bacteria identified substantially more frequently from afflicted than control eyes were Staphylococcus aureus, viridans Streptococci, and Eschlerichia coli, suggesting that these bacteria may be the cause of conjunctivitis. Chlamydiae, M. hominis, Neisseria gonorrhoeae, and anaerobic bacteria were all found to be negative in all cultures. The mother's race, socioeconomic position, sickness, or obstetric events had no influence on the frequency of conjunctivitis, the time it took to appear, or the microorganisms identified.[8]

Vasan et al (2020), They propose a new classifier based on CNN-based deep learning model to detect different versions of malware families and improve malware detection in this paper. Their method converts raw malware binaries into color images, which the fine-tuned CNN model uses to detect and identify malware families. They compare the IMCFN individual's performance to that of previous malware classification studies that used image-based malware classification techniques based on machine and deep learning methods.[9]

Humayun et al (2020), In this paper they present the results of a systematic mapping study that was undertaken to identify and analyse the common cyber security vulnerabilities. They divided their work into two part one was Cyber security and 2nd was Attack/threat/vulnerability. After that they identify and analyse the common cyber security

vulnerabilities. To achieve this goal, a systematic mapping study was conducted, and in total, 78 primary studies were identified and analysed.[10]

Chen et al (2020), In this paper, they proposed proposed an automatic vulnerability classification approach using the term frequency-inverse gravity mo- ment. The framework involves evaluation and classification. In the evaluation phase, different machine learning algorithms are eval- uated on ten vulnerable software applications.[11]

Kalash et al (2018), Here in this research, they defined that novel attack algorithm that generates adversarial malware binaries by only changing few tens of bytes in the le header. That's why they proposed that the other state-of-the-art attack algorithms, their attack did not require injecting any padding bytes at the end of the le, and it was much more efficient, as it requires manipulating much fewer bytes. After the analysis the malware in binaries they shown that, despite the success of deep learning in many areas, there are still uncertainties regarding the precision of the output of these techniques.[12]

Vinayakumar et al (2019), The authors provided a complete comparison analysis of their approach, which shows that their proposed models frameworks outmatch traditional MLAs. Their innovation in combining visualization and deep learning architectures for a modular system based on static, dynamic, and image processing implemented in a hadoop platform is the first of its kind in terms of achieving robust smart minimal malware detection. Their deep learning parameters were set using a hyper parameter selection approach with various trials of studies lasting up to 1,000 epochs and learning rates ranging from 0.01-0.5.[13]

Venkatraman et al (2019), They aim to do two things in this paper: 1. display the use of image-based techniques for detecting strange activities in systems, and 2. suggest and explore the use of hybrid image-based approaches with deep learning architectures for beneficial malware classification. Various similarity measures of malware behavioural traits, and also cost-sensitive deep learning architectures, are used to evaluate. They used eight different distance measures to determine the commonalities between malware variants, produces comparison vectors, including using images of the range scores to recognize the malware group.[14]

Jerbi et al (2020), The Artificial Malware-based Detection Approach (AMD) was investigated in this work as a new approach for Android malware detection. AMD is based on the creation of illusionary patterns that can detect malware using an iterative algorithm. The API call sequences extraction, pattern construction, and classification algorithm are the three key parts of the presented AMD strategy. FP = 00.21 percent and FN = 00.41 percent for the balanced data set, which analyzed 14 775 684 risks and 10 172 203 benign patterns, and FP = 00.28

percent and FN = 00.44 percent for the imbalanced data set, which evaluated 27 534 880 malicious patterns and 10 172 203 innocuous variations.[15]

Saad et al (2019), They explain how latest configuration machine learning techniques face particular challenges when detecting malware in the wild. They identified three key issues that limit the success of machine learning-powered malware detection systems in the wild. They then go over potential solutions to complex problems as well as the criteria for next-generation malware detection. The simulation results show that their FDP strategy is efficient, demonstrating that FDP can detect more Android malicious families than existing techniques.[16]

Jiang et al (2020), They propose a methodology for identifying Android malicious apps called fine-grained dangerous permission (FDP), which gathers features that better represent the difference between legitimate and malicious apps. The fine-grained feature of dangerous permissions applied in components is proposed for the first time among these characteristics. They also tested 1700 benign and 1600 malicious applications, demonstrating that FDP achieves a 94.5 percent TP rate. In addition, when compared to other related detection methods.[17]

Dovom et al (2019), In this paper, they performed a high level of accuracy in a decent amount of time, and even more so for the fast fuzzy pattern tree. Also used comprehensive segmentation and fuzzy classification, way that results in a stronger malware detection and classification approach for iot technology. following the implementation of their required fuzzy pattern with 275 different versions The fuzzy pattern tree might accurately identify malware and innocuous for the IoT and Vx-Heaven datasets with an accuracy of 99.834 percent and 100%, respectively, experimental measurements.[18]

Nix, R., & Zhang, J. (2017). They focus on strategy analysis in this paper, which explores Android system-API calls made by an app. System-API calls describe how an app expresses with the Android operating system. They establish a Convolutional Neural Network (CNN) for sequence classification and run a number of tests on malware detection and software classification into functionality groups to make a comparison our CNN with recurrent neural networks (LSTM) and other n-gram based techniques.[19]

Edmar Rezende et al. (2017) proposed a CNN-based architecture for malware classification. This method used the pre-training models of VGG16 and achieved high accuracy. Similarly, also used transfer learning to apply the ResNet-50 architecture for malware classification. The deep neural network was trained by freezing the convolutional layers of ResNet-50 pre-trained

on the ImageNet dataset. Motivated by the visual similarity between malware from the same family. [20]

Liu et al. (2017) offered a step-by-step method for automatically classifying malware into families and detecting new infections. Grey-scale byte plots, Op-code n-grams, and import functions were all employed. These features are used by the decision-making module to classify malware samples to their own families, as well as to find novel viruses that have not been seen before. They made use of Shared. The clustering technique Nearest Neighbour (SNN) was used to discover new malware families. Their approach is tested on a dataset of 21,740 malware samples 98.9% classification accuracy was reported using samples from nine different families.[21]

Zhang et al (2019), In this paper, they define that they are the first to suggest a method for classifying malware based on static analysis. First, ransomware samples' opcode patterns are converted to N-gram patterns. To validate the model, they employ six assessment methods. Experiment results using real datasets show that the method achieves the highest accuracy of 91.43 percent. Furthermore, the "want tocry" malware family's average F1-measure is up to 99 percent, and 2-class accuracy is up to 99.3 percent. They also mentioned that they observed the various feature measurements needed to achieve similar classifier performance with feature N-grams of various lengths.[22]

Ye et al (2017), They give a general summary of malware and the anti-malware manufacturers, and also the essential commodities for malware detection and intelligent malware detection methods, in this article. The process of detection is usually divided into two stages in these strategies: feature extraction and categorization. They also go through some of the other problems and concerns of malware detection using data mining methods, as well as forecasting malware emerging trends.[23]

Kumar et al. (2019) Using a Random Forest classifier, researchers employed a combination of static and dynamic algorithms to categorize malware into classes in the first 4 seconds of runtime. Early-stage detection refers to stopping a procedure before it has completed its analysis. They collected information from the PE header, such as the file header, optional header, and section header, using static analysis. They also retrieved data from the section table and sections, such as the number of sections, their size, and the virtual address of each section, among other things. They collected features based on important resources such as network data, system calls, processes, and registry from dynamic analysis. Following that, the Information Gain method is used to decrease the feature set. Finally, the generated feature vector is used by

training for classification. Random Forest, Decision Tree, XGBoost, Neural Network, and K-NN classifiers are among the classifiers used.[24]

Moshiri et al (2017) use Features are extracted from the JSON reports as Source of Malware and Mutation Information (MI) as feature selection technique and MLP as a detection technique and get 97% accuracy and it identifies the obscure malware by registering its comparative known malware profile.[25]

Raff et al (2018), In this paper, They use times larger more data in this study, and we use Elastic-Net regularized Logistic Regression to perform feature selection during model building. They also describe novel multi-byte identifiers and compute a regularization path. They also show that, even with linear models and extraordinary regularization, n-gram features promote overfitting. The features chosen first in the path are those that the model found to be the most predictive of the 200k beginning set.[26]

Pendlebury et al (2019), They present a new metric that summarizes a classifier's expected robustness in a real-world setting, as well as a technique to tune its performance, in this paper. They also show how this can be used to evaluate time decay mitigation techniques like effective learning. They said they used TESSERACT, an open-source evaluation framework for making comparisons malware classifiers in a realistic setting, to implement their solutions.[27]

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Methodology Model

I am applying different kinds of machine learning models for detecting malware. Here in this chapter, I will be defining the process of my collecting dataset and also define the model's definition those are using this dataset and classify the malware. in this diagram [Fig: 1] which is given the clear concept in my hole research. And also, able to understand how I work in my research and which stap I flow from start to end.

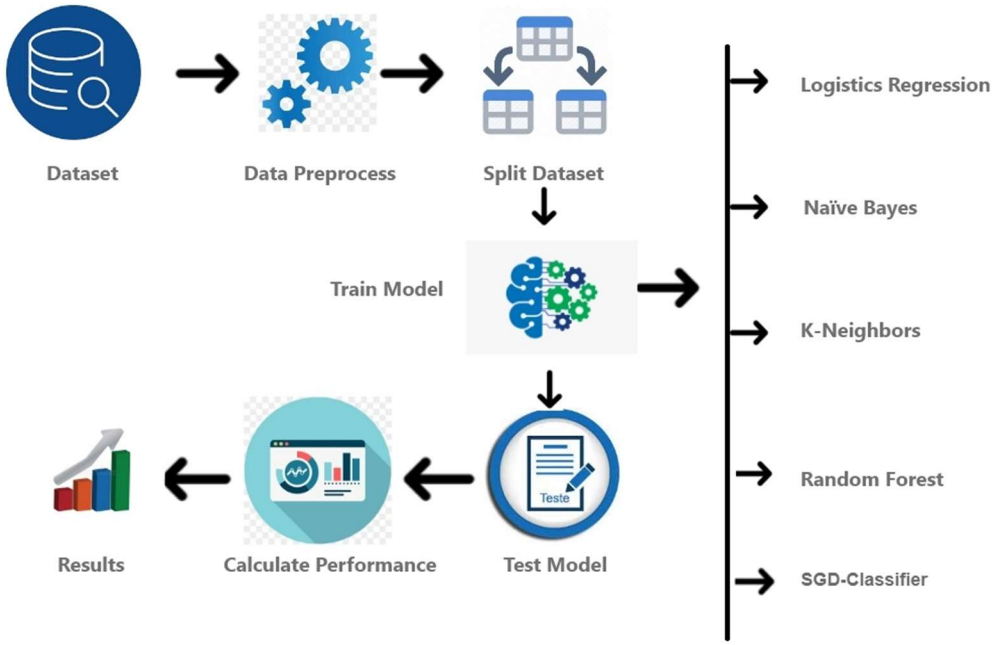


Fig 1: Methodology Model

3.2 Data pre-processing

In my dataset I also declared that, this dataset has large number of training set. That's why it was very difficult to prepossess this dataset. Here in this data set I am show that how many columns have any null value in the null value columns I am filling those null row with most frequent number in this column. After that, I am implementing this dataset feature engineering and also feature extraction. Then also implement their min-max scalar for reduce unimportant features. After that, complete all staps [Fig 2:] shows that my feature those are used detect the correlation matrix for model for detecting the malware.

Fig 2: Selected features

3.3 Data Visualization

Here in the [Fig: 3] according to the all features the target variable which is legitimate or not its define that how many 0 and how many 1 in this dataset. That's a ratio diagram for all data set which is define how many rows is 0 and how many row is 1 where 0 means not legitimate and 1 means legitimate which is define a malware.

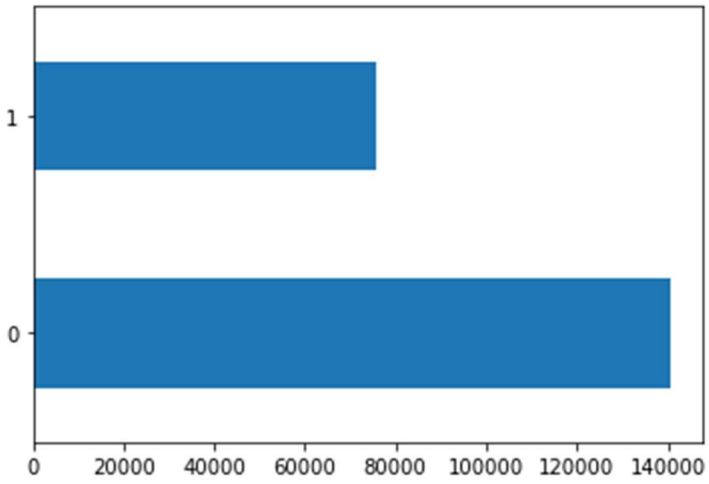


Fig 3: Bar diagram of Target variable

For better understand the data set here [Fig: 4] shown the histogram of some sample features. Which is clearly define that all data features in this dataset have some integers value and also have some categorical value. And all features in this dataset are integer value and it's a numerical data.

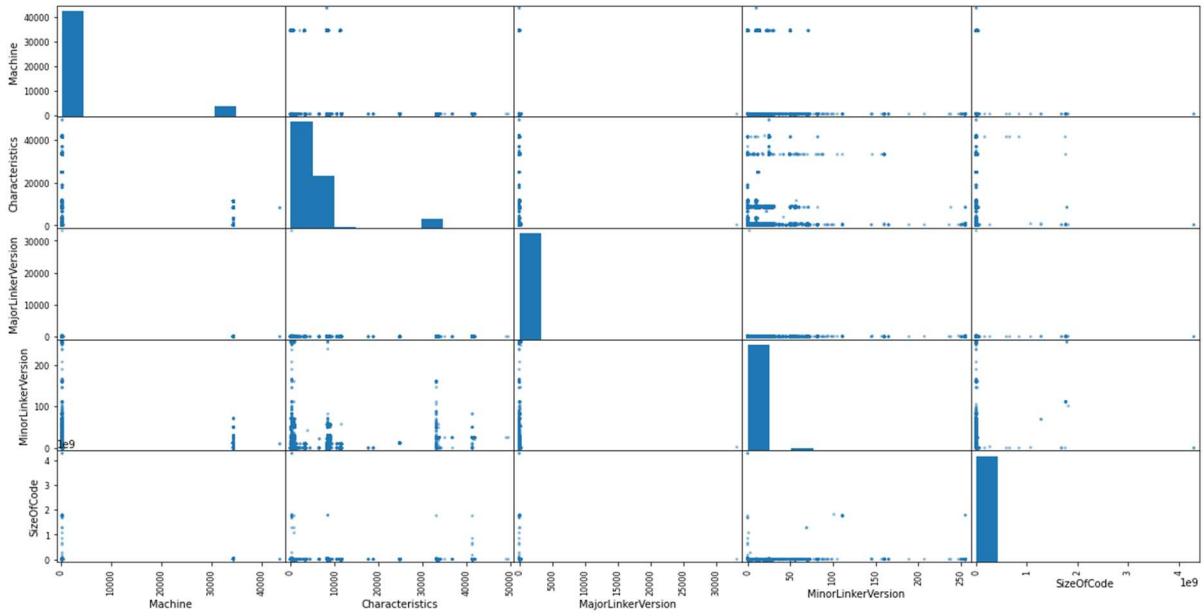


Fig 4: Sample features Histogram

3.4 feature engineering

Feature engineering is a machine learning technique for producing new variables that were not in the training set using data.[31] It can generate new features for both supervised and unsupervised learning, with the goal of making data transformations simpler and faster while also improving model accuracy. When working with machine learning models, feature engineering is required. A bad feature, regardless of the data or architecture, will have a major effect on the Machine Learning model.

3.5 Normalization

Normalization is a data pre-processing method that is frequently used in machine learning. Normalization is the process of transforming the qualities of numeric columns in a dataset to a common scale without perverting the ranges of values or losing information [31]. Through normalisation, all values are scaled in a specified range between 0 and 1. (or min-max normalisation). This change has no impact on the extent of the feature, but it does amplify the effects of outliers due to lower standard deviations. As a result, dealing with outliers before normalization is recommended.

3.6 Correlation

I found the correlation in this dataset for all features which is related to detect the malware. After showing the all-features correlation value here I am implemented correlation matrix function for reducing the thresholds value which is relatable above 50% in this dataset features

those are given for build the models. And under 50% of thresholds, we delete those features with this function. [Fig: 5] Show the correlation plot those are accepted for building models.

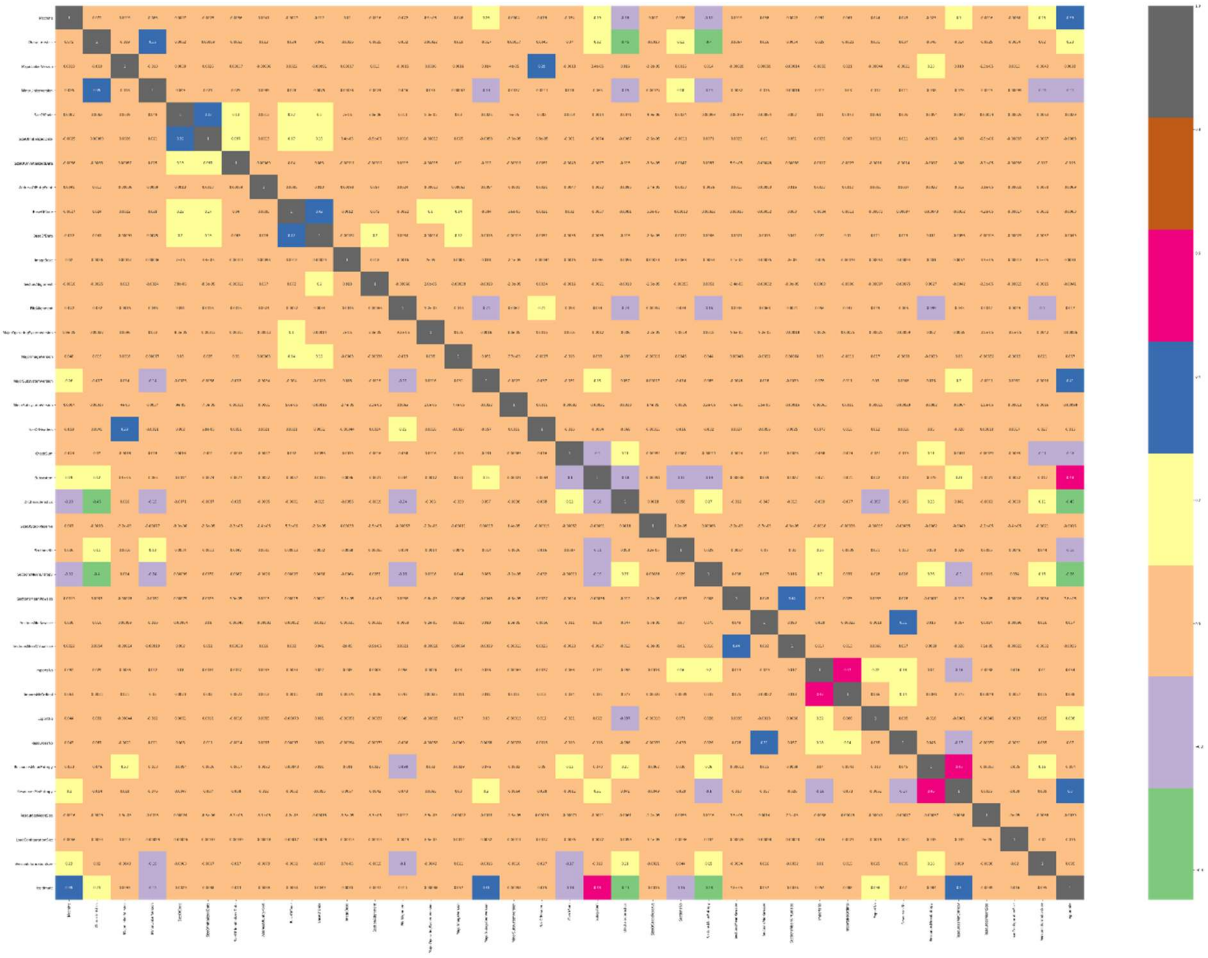


Fig 5: Correlation Plot for Most relatable features

3.7 Logistics Regression

Although the name "logistic" seems perplexing, you can rest certain that it is not a regression algorithm. Logistic The term "regression" refers to a type of classification. It predicts discrete values such as true or false, yes or no, 0 or 1, and so on, based on a set of independent factors (s). In other words, it estimates the chance of any event occurring based on the data. Because it guesses the probability, the results are always erroneous between 0 and 1.

$$g(E(y)) = \alpha + \beta x1 + \gamma x2 \qquad \text{(Equation: 1)}$$

Here, g() is the link function, E(y) is the expectation of target variable and $\alpha + \beta x1 + \gamma x2$ is the linear predictor (α,β,γ to be predicted). The role of link function is to 'link' the expectation of y to linear predictor.

3.8 Naïve Bayes

Naïve Bayes is a form of probabilistic classifier that is simple. The name nave denotes that it foresees features that will result in an independent model. It is based on the Bayes theorem and uses strong (nave) data. It is a collection of Bayes theorem-based classification methods. Independence assumptions are one of the aspects. It is discovered that the presence of a given feature in a class is unrelated to the presence of any other characteristics.

3.9 K-Neighbors Classifier

The K-Neighbors method is one of the most fundamental Machine Learning algorithms and is based on the Supervised Learning approach. This approach assumes that the new case/data and previous cases are comparable and assigns the new case to the category that is the most similar to the existing categories. The K- K-Neighbors approach saves all available data and classifies new data points based on their similarity to current data. This means that utilizing this approach, fresh data may be swiftly sorted into a well-defined category. Although the this technique may be used for both regression and classification, it is more typically employed for classification problems.

3.10 Random Forest

This is a strategy that can be used for both classification and regression. A forest is made up of trees that grow together to form a woodland or jungle. Random Forest analysis most likely creates decision trees based on data sets and then extracts values from each of them. Finally, by means of voting, selects the best layout. It's a manner of arranging things. It performs far better than a single decision tree. Because it averages the outcome, it reduces over-fitting.

3.11 SGD Classifier

SGD (Stochastic Gradient Descent) is a simple yet effective optimization method for identifying the values of function parameters/coefficients that minimize a cost function. To put it another way, it's used to train discriminative linear classifiers with convex loss functions, such as SVM and Logistic regression. It has been effectively used to large-scale datasets since the update to the coefficients is performed for each training instance rather than at the end of instances. The Stochastic Gradient Descent (SGD) classifier is just a standard SGD learning approach that supports a variety of loss functions and classification penalties. Scikit-learn includes the SGD Classifier module, which may be used to do SGD classification.

3.12 Performance Calculation

For each class, we examine Accuracy score, Precision value, F1 Score, Recall, and Support in order to select the best model for our proposed system "(1-4)".

1. Precision = $\frac{True\ positive}{True\ Positive + False\ Positive}$

2. Recall = $\frac{True\ positive}{True\ Positive + Fals\ \ Negative}$

3. F1 value = $\frac{True\ Positive}{True\ Positive + \frac{Zero\ Negative \times Zero\ Positive}{2}}$

4. Accuracy=$\frac{predictions\ of\ correct}{Predictions\ of\ all}$

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The demonstration's overall preparation is decomposed into several steps, including dataset collection, level generation, data resizing, proposed model portrayal, and model preparation strategy.

4.2 Analysis Technique

In a jupyter notebook browser application, I displayed a technology based on the Python programming language. For the described language, library, and visualization tools, I am using Python 3.9 version. Those are all integrated into jupyter notebook, an open-source and conda application. Each cell in a document contains the script or markdown code language, which is embedded in the final content. Typical outputs include features, tables, charts, and graphs. Because tests and results are presented in a self-contained format, this technology makes it simple to exchange and copy scientific works. The jupyer notebook is a project dedicated to promoting machine learning education and research. Jupyter notebooks work similarly to Microsoft Office objects in that they can be shared and multiple users can work on the same notebook at the same time. Pandas, numpy, seaborn, matplotlib, and sklearn are just a few of the pre-installed machine learning and AI libraries in the Python 2 and 3 runtimes. After a period of time under runtime (VM), the virtual machine becomes dormant, and all user data and configuration is lost. The notebook, on the other hand, is secure, and files can be transferred from the VM hard disk to the user's Daffodil international University Google Drive account. Finally, this jupyter notebook is a GPU-accelerated runtime that is fully integrated with the technologies previously discussed.

4.3 Labels generation

This dataset was divided into two category those are malware for 1 and not a malware for 0 in the train data. In the bellow [Fig: 6] define the level of my datasets. Here we can clearly see that there have above 50% of data are 1 that’s mean in this data set have malware features and other feature value define that those are not a malware software that’s mean those are safe software for all users. This train dataset defines that which specific features value belongs a malware and which is not.

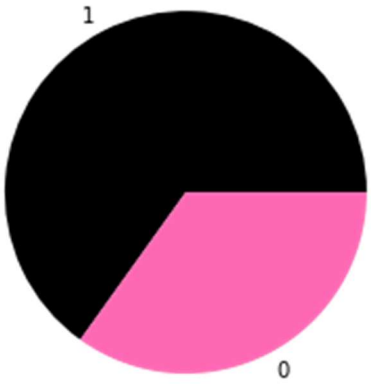


Fig 6: Level for classification dataset

4.4 Training process

About 90% data form my dataset I am using as a train data and other 10% data I am using for test my models for showing the model accurateness for this dataset. Those validation result give a clear idea that how my model work in this dataset and how perfect it's prediction.

4.5 Models Result

In this section I am discus about my models result after that I will decide which model is the best model for the malware detection.

Models name	Train Accuracy	Test Accuracy	Model Validation
Logistic Regression	0.883	0.885	**Overfit (Implement k-fold Cross Validation)* **Accuracy: 0.935**
Naive Bayes Classifier	0.911	0.909	Ok
K-Neighbors Classifier	0.999	0.973	Ok
Random Forest Classifier	0.998	0.987	Ok
SGD Classifier	0.857	0.854	Ok

Tab 1: Accuracy Table

Here we can see [Tab:1] that Random Forest Classifier has highest accuracy which is almost 99% and there is no doubt that this model is one of the best models from the other classification models. There for I am declared that this model is the best for detecting malware. And also, this model is the more significant model from other for detecting the malware according to

those all-usable features. Also can visualize the model accuracy with both train and test and [Fig: 7] clearly define show that Random Forest Classifier has best accuracy of all result in this bar diagram.

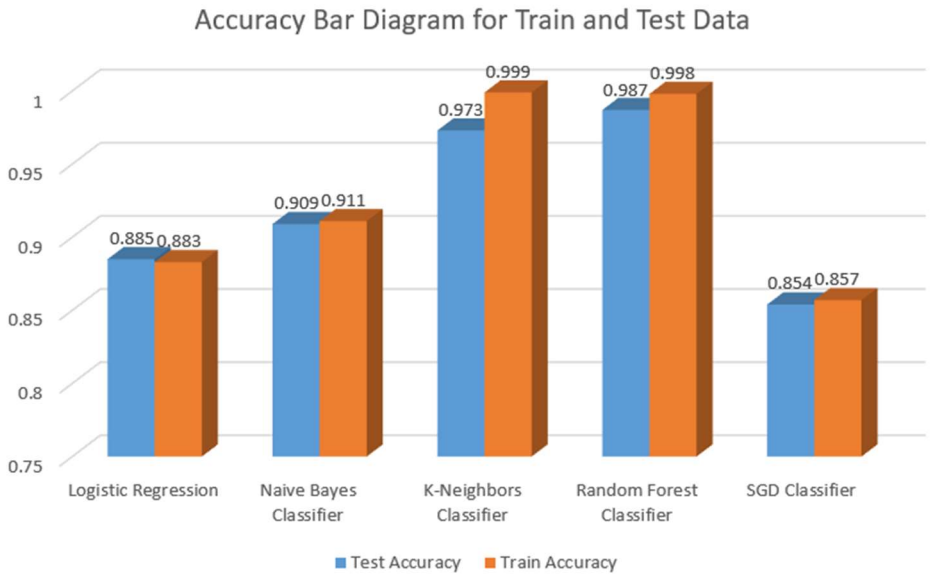


Fig 7: Accuracy result bar diagram

4.6 Model evaluation

In this section now I am discus how accurate for detecting the malware. All evaluation result given bellow which is found after train all of the given models:

Model name	Precision	Recall	f1-score	Class
Logistic Regression	0.89	0.95	0.92	0
	0.88	0.77	0.82	1
Naïve Bayes Classifier	0.93	0.93	0.93	0
	0.87	0.87	0.87	1
K-Neighbors Classifier	0.98	0.98	0.98	0
	0.96	0.96	0.96	1
Random Forest Classifier	0.99	0.99	0.99	0
	0.98	0.98	0.98	1
SGD Classifier	0.96	0.81	0.88	0
	0.73	0.93	0.82	1

Tab 2: Performance Table

Here we can the all precision, recall and fi-scores [Tab:2] for each class for detecting the malware from the test dataset. Here also clearly define that Random Forest Classifier is the more accurate from the other models. Here in this model, I have 99% precision and 99% recall for the malware class and 98% precision with 98% recall for the not malware class. That why this is the best model with almost 99% accuracy [Tab:2] for detecting the malware. Here also I am defining the [Fig: 8-12] truth table diagram for all models for better understand the [Tab:2].

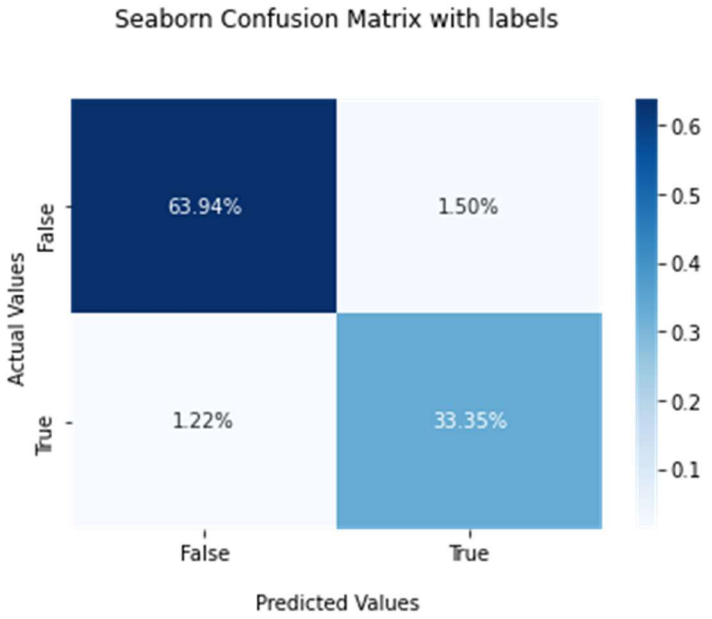


Fig 8: Truth Table Diagram for K-Neighbors Classifier

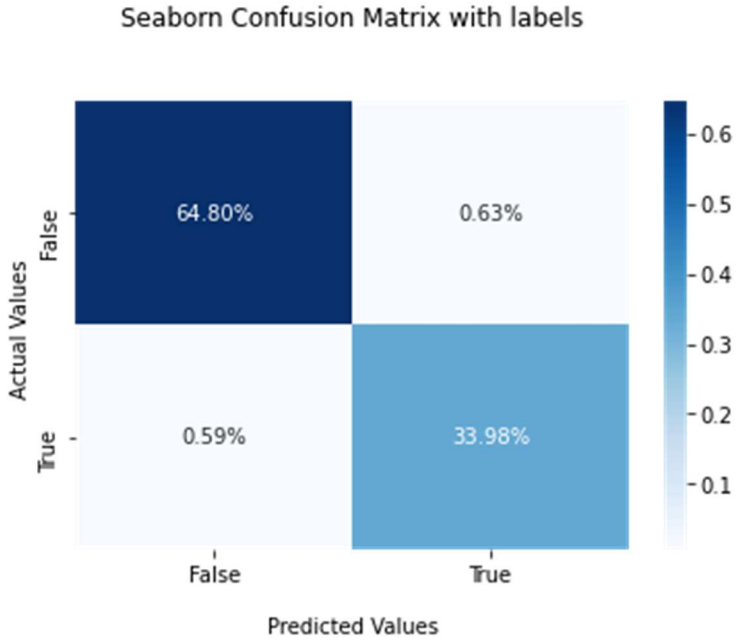


Fig 9: Truth Table Diagram for Random Forest Classifier (Best Model)

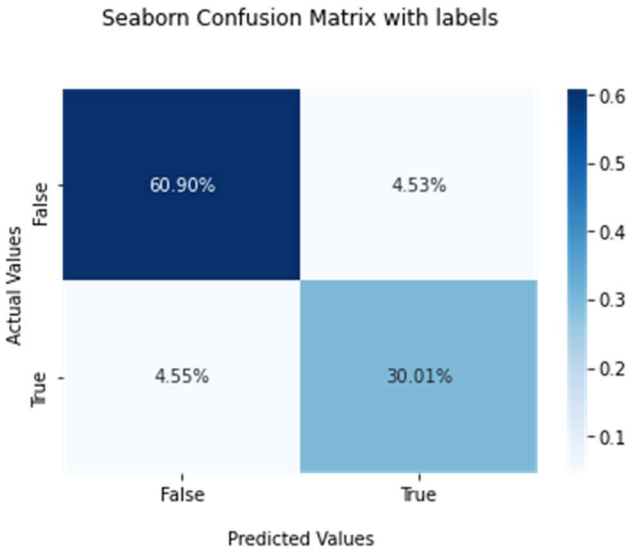


Fig 10: Truth Table of Naïve Bayes Classifier

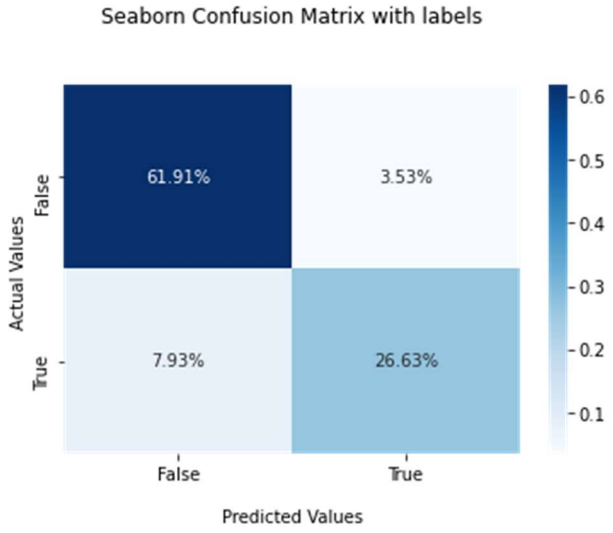


Fig 11: Truth Table of Logistic Regression

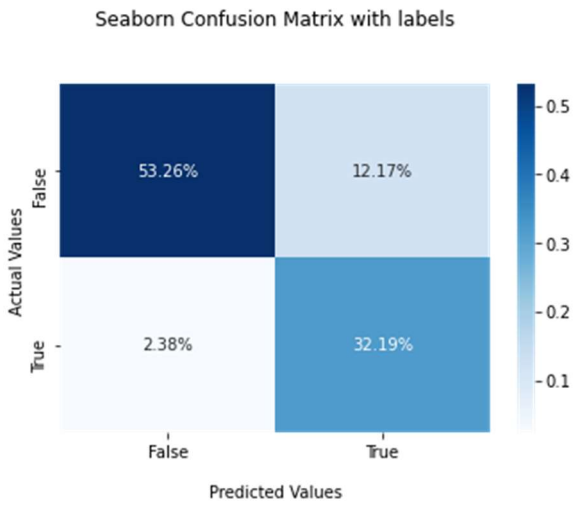


Fig 12: Truth Table of SGD Classifier

4.7 Model Validation ROC diagram

ROC Diagram

We also now know that the AUC-ROC curve aids in visualizing how well our machine learning classifier performs. Although it only works for binary classification problems, we will see [Fig 13-17] how we can extend it to evaluate multi-class classification problems in the end.

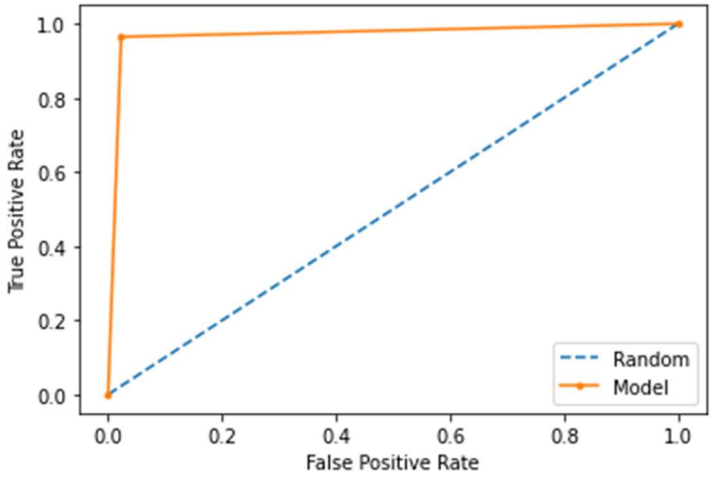


Fig 13: ROC curve of K-Neighbors Classifier

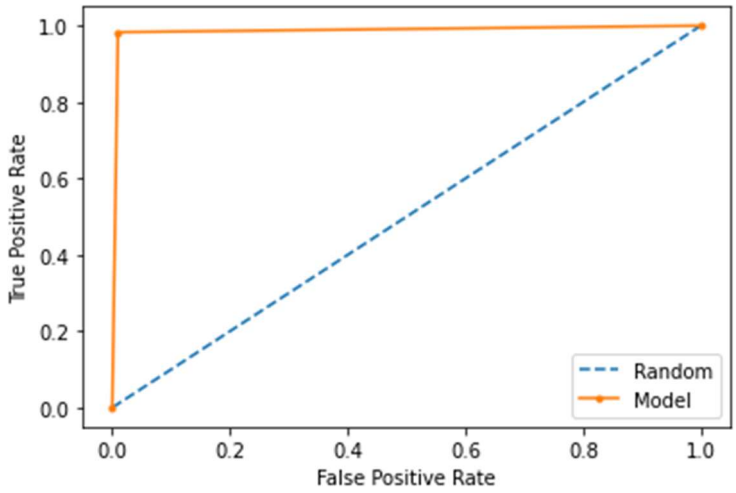


Fig 14: ROC curve of Random Forest Classifier (Best Model)

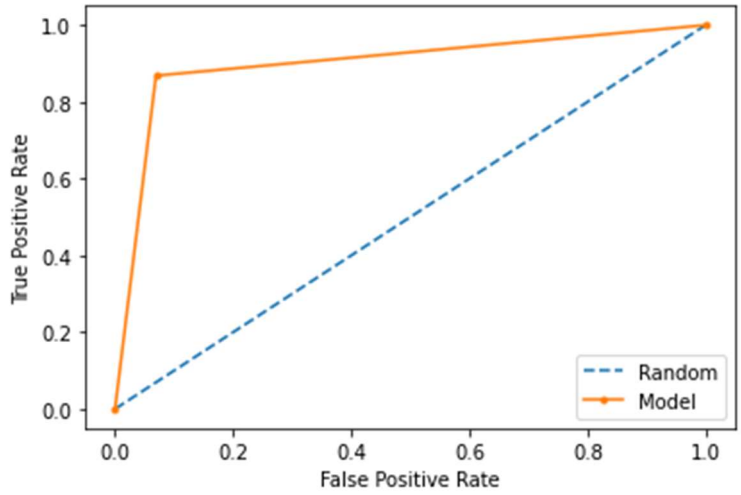


Fig 15: ROC curve of Naïve Bayes Classifier

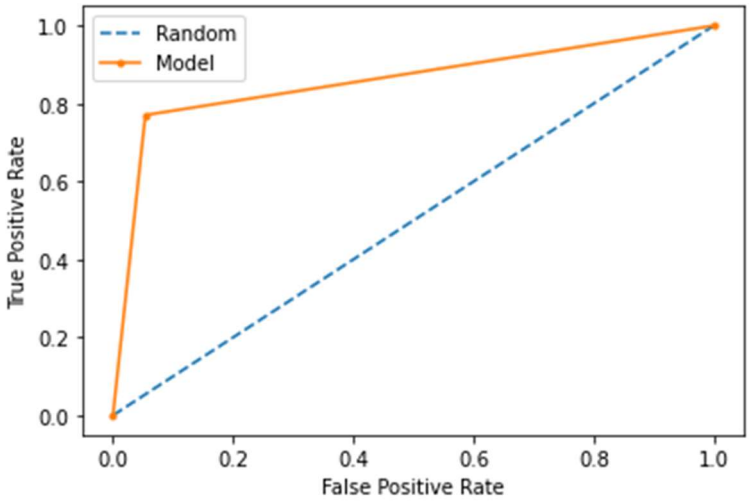


Fig 16: ROC curve of Logistic Regression

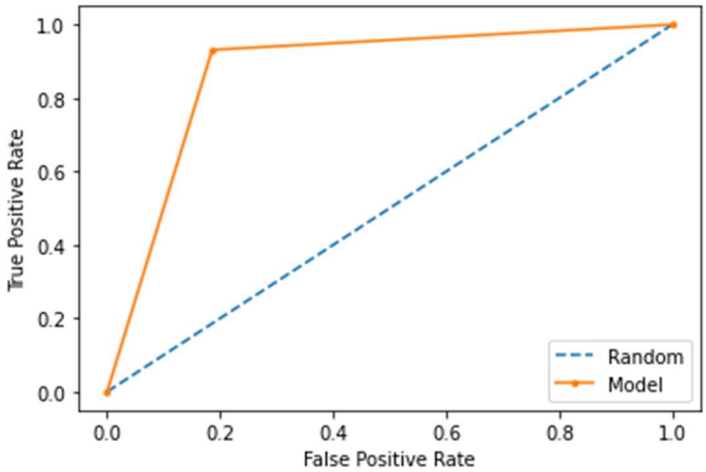


Fig 17: ROC curve of SGD Classifier

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Malware classification is crucial for determining the origins of computer security attacks. Existing static analysis tools, on the other hand, are quick in classification but ineffective in some malware that use packing and obfuscation tactics. Machine learning technology was used to detect malware in this study. A high-precision malware detection method was proposed based on five machine learning classification models. First, a malware dataset was gathered, which included two types of classes: malware and non-malware. There is no doubt that a great deal of research has committed to going into identifying malware. The major purpose of this study was to introduce the malware classification system, which was based on machine learning. After reviewing a large number of papers, we use four categorization techniques. Random Forest Classifier outperforms them all, with almost 99 percent accuracy rate. Second, K-Neighbors Classifier comes in second place in terms of malware classification accuracy, with a score of 97 percent [Tab:1].

5.2 Future Work

In future, I will continue to optimize those classification models and also want to implement the deep learning classification model algorithm. Also, I want to implement time complexity to find the detection time for any kind of malware. Also want to implement this model as a malware detection software which will be use in any kind of device.

REFERENCES

[1] Xue, D., Li, J., Lv, T., Wu, W., & Wang, J. (2019). Malware classification using probability scoring and machine learning. IEEE Access, 7, 91641-91656.

[2] Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018, February). Malware classification with deep convolutional neural networks. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

[3] Milosevic, N., Dehghantanha, A., & Choo, K. K. R. (2017). Machine learning aided Android malware classification. Computers & Electrical Engineering, 61, 266-274.

[4] Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. Computers & Security, 81, 123-147.

[5] Fu, J., Xue, J., Wang, Y., Liu, Z., & Shan, C. (2018). Malware visualization for fine-grained classification. IEEE Access, 6, 14510-14523.

[6] Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. Journal of Network and Computer Applications, 153, 102526.

[7] Zhang, Y., Huang, Q., Ma, X., Yang, Z., & Jiang, J. (2016, August). Using multi-features and ensemble learning method for imbalanced malware classification. In 2016 IEEE Trustcom/BigDataSE/ISPA (pp. 965-973). IEEE.

[8] Kang, B., Yerima, S. Y., McLaughlin, K., & Sezer, S. (2016, June). N-opcode analysis for android malware classification and categorization. In 2016 International conference on cyber security and protection of digital services (cyber security) (pp. 1-7). IEEE.

[9] Vasan, D., Alazab, M., Wassan, S., Naeem, H., Safaei, B., & Zheng, Q. (2020). IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. Computer Networks, 171, 107138.

[10] Humayun, M., Niazi, M., Jhanjhi, N. Z., Alshayeb, M., & Mahmood, S. (2020). Cyber security threats and vulnerabilities: a systematic mapping study. Arabian Journal for Science and Engineering, 45(4), 3171-3189.

[11] Chen, J., Kudjo, P. K., Mensah, S., Brown, S. A., & Akorfu, G. (2020). An automatic software vulnerability classification framework using term frequency-inverse gravity moment and feature selection. Journal of Systems and Software, 167, 110616.

[12] Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018, February). Malware classification with deep convolutional neural networks. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

[13] Vinayakumar, R., Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, and Sitalakshmi Venkatraman. "Robust intelligent malware detection using deep learning." IEEE Access 7 (2019): 46717-46738.

[14] Venkatraman, S., Alazab, M., & Vinayakumar, R. (2019). A hybrid deep learning image-based analysis for effective malware detection. Journal of Information Security and Applications, 47, 377-389.

[15] Jerbi, M., Dagdia, Z. C., Bechikh, S., & Said, L. B. (2020). On the use of artificial malicious patterns for android malware detection. Computers & Security, 92, 101743.

[16] Saad, S., Briguglio, W., & Elmiligi, H. (2019). The curious case of machine learning in malware detection. arXiv preprint arXiv:1905.07573.

[17] Jiang, X., Mao, B., Guan, J., & Huang, X. (2020). Android malware detection using fine-grained features. Scientific Programming, 2020.

[18] Dovom, E. M., Azmoodeh, A., Dehghantanha, A., Newton, D. E., Parizi, R. M., & Karimipour, H. (2019). Fuzzy pattern tree for edge malware detection and categorization in IoT. Journal of Systems Architecture, 97, 1-7.

[19] Nix, R., & Zhang, J. (2017, May). Classification of Android apps and malware using deep neural networks. In 2017 International joint conference on neural networks (IJCNN) (pp. 1871-1878). IEEE.

[20] Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., & De Geus, P. (2017, December). Malicious software classification using transfer learning of resnet-50 deep neural network. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1011-1014). IEEE.

[21] Liu, L., Wang, B. S., Yu, B., & Zhong, Q. X. (2017). Automatic malware classification and new malware detection using machine learning. Frontiers of Information Technology & Electronic Engineering, 18(9), 1336-1347.

[22] Zhang, H., Xiao, X., Mercaldo, F., Ni, S., Martinelli, F., & Sangaiah, A. K. (2019). Classification of ransomware families with machine learning based on N-gram of opcodes. Future Generation Computer Systems, 90, 211-221.

[23] Ye, Y., Li, T., Adjeroh, D., & Iyengar, S. S. (2017). A survey on malware detection using data mining techniques. ACM Computing Surveys (CSUR), 50(3), 1-40.

[24] Kumar, N., Mukhopadhyay, S., Gupta, M., Handa, A., & Shukla, S. K. (2019, August). Malware classification using early stage behavioral analysis. In 2019 14th Asia Joint Conference on Information Security (AsiaJCIS) (pp. 16-23). IEEE.

[25] Moshiri, E., Abdullah, A. B., Mahmood, R. A. B. R., & Muda, Z. (2017). Malware Classification Framework for Dynamic Analysis using Information Theory. Indian Journal of Science and Technology, 10(21), 1-10.

[26] Raff, E., Zak, R., Cox, R., Sylvester, J., Yacci, P., Ward, R., ... & Nicholas, C. (2018). An investigation of byte n-gram features for malware classification. Journal of Computer Virology and Hacking Techniques, 14(1), 1-20.

[27] Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., & Cavallaro, L. (2019). {TESSERACT}: Eliminating experimental bias in malware classification across space and time. In 28th {USENIX} Security Symposium ({USENIX} Security 19) (pp. 729-746).

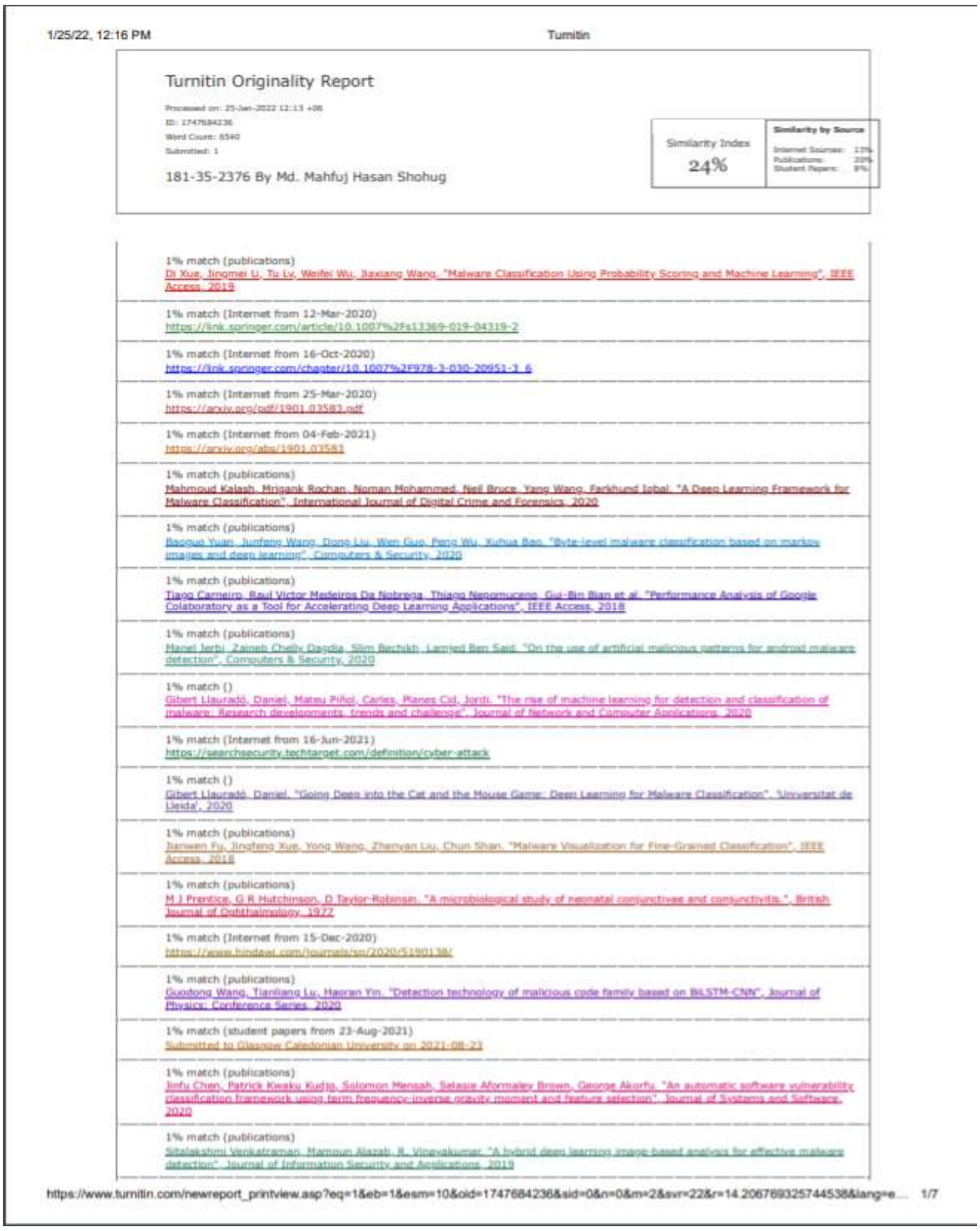
[28] AO Kaspersky Lab. Kaspersky Security Bulletin Overall Statistics for 2017. Accessed: Jul. 22, 2018. [Online]. Available: https://securelist. com/ksb-overall-statistics-2017/83453/

[29] [B. Christiaan, D. Taylor, G. Steve, K. Mary, M. Niamh, and P. Chris. (Jun. 2018). McAfee Labs Threats Reports. McAfee. Accessed: Aug. 2, 2018. [Online]. Available: https://www.mcafee.com/ enterprise/en-us/assets/reports/rp-quarterly-threats-jun-2018.pdf

[30] L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang, ``A comparative assessment of malware classification using binary texture analysis and dynamic analysis," in Proc. ACM Workshop Secur. Artif. Intell., 2011, pp. 21-30.

[31] L. Nataraj, S. Karthikeyan, and B. Manjunath, “Sattva: Sparsity inspired classification of malware variants,” in Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security. ACM, 2015, pp. 135–140.

PLAGIARISM REPORT



1/25/22, 12:16 PM Turnitin

Turnitin Originality Report

Processed on: 25-Jan-2022 12:13 +06
ID: 1747684236
Word Count: 6540
Submitted: 1

181-35-2376 By Md. Mahfuj Hasan Shohug

Similarity Index
24%

Similarity by Source
Internet Sources: 13%
Publications: 20%
Student Papers: 8%

1% match (publications)
Di Xue, Jingmei Li, Tu Lv, Weifei Wu, Jiaxiang Wang, "Malware Classification Using Probability Scoring and Machine Learning", IEEE Access, 2019

1% match (Internet from 12-Mar-2020)
https://link.springer.com/article/10.1007%2Fs13369-019-04319-2

1% match (Internet from 16-Oct-2020)
https://link.springer.com/chapter/10.1007%2F978-3-030-20951-3_6

1% match (Internet from 25-Mar-2020)
https://arxiv.org/pdf/1901.03583.pdf

1% match (Internet from 04-Feb-2021)
https://arxiv.org/abs/1901.03583

1% match (publications)
Mahmoud Kalash, Mrigank Rochan, Noman Mohammed, Neil Bruce, Yang Wang, Farkhund Iqbal, "A Deep Learning Framework for Malware Classification", International Journal of Digital Crime and Forensics, 2020

1% match (publications)
Baoguo Yuan, Junfeng Wang, Dong Liu, Wen Guo, Peng Wu, Xuhua Bao, "Byte-level malware classification based on markov images and deep learning", Computers & Security, 2020

1% match (publications)
Tiago Carneiro, Raul Victor Medeiros Da Nobrega, Thiago Nepomuceno, Gui-Bin Bian et al, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications", IEEE Access, 2018

1% match (publications)
Manel Jerbi, Zaineb Chelly Dagdia, Slim Bechikh, Lamjed Ben Said, "On the use of artificial malicious patterns for android malware detection", Computers & Security, 2020

1% match ()
Gibert Llauradó, Daniel, Mateu Piñol, Carles, Planes Cid, Jordi, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenge", Journal of Network and Computer Applications, 2020

1% match (Internet from 16-Jun-2021)
https://searchsecurity.techtarget.com/definition/cyber-attack

1% match ()
Gibert Llauradó, Daniel, "Going Deep into the Cat and the Mouse Game: Deep Learning for Malware Classification", 'Universitat de Lleida', 2020

1% match (publications)
Jianwen Fu, Jingfeng Xue, Yong Wang, Zhenyan Liu, Chun Shan, "Malware Visualization for Fine-Grained Classification", IEEE Access, 2018

1% match (publications)
M J Prentice, G R Hutchinson, D Taylor-Robinson, "A microbiological study of neonatal conjunctivae and conjunctivitis.", British Journal of Ophthalmology, 1977

1% match (Internet from 15-Dec-2020)
https://www.hindawi.com/journals/sp/2020/5190138/

1% match (publications)
Guodong Wang, Tianliang Lu, Haoran Yin, "Detection technology of malicious code family based on BiLSTM-CNN", Journal of Physics: Conference Series, 2020

1% match (student papers from 23-Aug-2021)
Submitted to Glasgow Caledonian University on 2021-08-23

1% match (publications)
Jinfu Chen, Patrick Kwaku Kudjo, Solomon Mensah, Selasie Aformaley Brown, George Akorfu, "An automatic software vulnerability classification framework using term frequency-inverse gravity moment and feature selection", Journal of Systems and Software, 2020

1% match (publications)
Sitalakshmi Venkatraman, Mamoun Alazab, R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection", Journal of Information Security and Applications, 2019

https://www.turnitin.com/newreport_printview.asp?eq=1&eb=1&esm=10&oid=1747684236&sid=0&n=0&m=2&svr=22&r=14.20678932574453&lang=e... 1/7