Spring 4-28-2020

Performance of Malware Classification on Machine Learning using Feature Selection

Nusrat Asrafi
Kennesaw State University

PERFORAMANCE OF MALWARE CLASSIFICATION ON COMPARNING DIFFERENT MACHINE LEARNING USING FEATURE SELECTION

A Dissertation
Presented to
The Academic Faculty

by

Nusrat Asrafi

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science

April 2020

PERFORAMANCE OF MALWARE CLASSIFICATION ON MACHINE LEARNING USING FEATURE SELECTION

Approved by:

Dr. Dan Chia-Tien Lo, Advisor
Department of Computer Science
Kennesaw State University

Dr. Donghyun Kim
Department of Computer Science
Kennesaw State University

Dr. Kai Qian
Department of Computer Science
Kennesaw State University

Date Approved: [March 28, 2020]

ACKNOWLEDGEMENTS

I would like to thank my thesis advisors Dr. Dan Chia-Tien Lo for his support and encouragement through this entire process. I am very thankful for this experience. I would also like to thank Dr. Kai Qian and Dr. Donghyun Kim, my thesis committee members for their insightful comments and valuable suggestions.

TABLE OF CONTENTS

CONTENTS

ACKNOWLEDGEMENTS **iv**

LIST OF TABLES Error! Bookmark not defined.

LIST OF FIGURES **vi**

LIST OF SYMBOLS AND ABBREVIATIONS Error! Bookmark not defined.

SUMMARY **viii**

Chapter 1 Introduction **1**
1.2 Motivation and Problem Statement **2**
1.3 Contribution of the research **4**
1.4 Organization of Research **5**
2.1 Overview **7**
2.2 Introduction **7**
2.3 Malware Classification **8**
2.3 Feature selection Methods for Malware Classification **10**
2.3.1 Types of Feature Selection 11
2.3.2 Previous Work on Feature Selection on Malware Detection 12
2.4 Malware classification in Machine learning **13**

Chapter 3 Feature Engineering **16**
3.1 Overview **16**
3.2 Description of dataset **16**
3.3 Dataset preprocessing **17**
3.3 Feature Selection **17**
3.3.1 No Feature Selection 18
3.3.2 FDR Feature Selection 19
3.3.3 Chi2-FDR Feature Selection 21
3.3.4 RFE Feature Selection 22
3.3.5 Lasso Regression: 23

Chapter 4 Machine Learning **26**
4.1 Overview **26**
4.2 SVM **26**
4.2.1 Linear Kernel 26
4.2.2 Poly Kernel 27
4.2.3 RBF Kernel 28
4.3 Random Forest **29**
4.4 MLP **30**

Chapter 5 Feature Importance **33**
5.1 Overview **33**
5.2 Feature importance Ranking using Random Forest Classifier **33**
5.3 Frequency Distribution of Top Feature **34**
5.3.1 The API Sequence length 34
5.3.2 Idgetprocedureaddress 36
5.3.3 ntclose 37
5.3.4 ntallocatevirtualmemory 39
5.3.5ldrloaddll 40

Chapter 6 Conclusion and Future Work **42**

6.1 Conclusion **42**
6.2 Future Work **42**

References **43**

LIST OF FIGURES

Figure 1.1 The growth of malwares in past 10 years 8
Figure 2 The growth of malware in past 10 years 17
Figure 3 ROC curve for SVM with RBF kernel using no feature selection 18
Figure 4 ROC curve for SVM with RBF kernel using 191 features 20
Figure 5 ROC curve for SVM with RBF kernel using 100 features 21
Figure 6 ROC curve of SVM with RBF kernel using Chi2 FDR selected top 100 features 22
Figure 7 ROC curve of SVM with RBF kernel using RFE Selected 100 Features 23
Figure 8 ROC curve of SVM with RBF kernel using LASSO Selected 100 Features 24
Figure 9 AUC of SVM with RBF kernel using ANOVA,ANOVA with FDR, Chi Square with FDR, RFE, Lasso Feature Selection Method with no feature selection 25
Figure 10 ROC curve of SVM with Linear Kernel using Anova FDR Selected 100 Features 27
Figure 11 ROC of SVM with Poly Kernel using Anova FDR Selected 100 Features 28
Figure 12 ROC curve of SVM with RBF kernel using FDR Selected 100 Features 29
Figure 13 ROC Curve of Random Forest using Anova FDR for top 100 features 30
Figure 14 ROC Curve of MLP Anova FDR Selected 100 Features 31
Figure 15 Comparison of AUC Curve in Linear SVM,SVM with Polynomial, SVM with RBF, Random Forest and Multilayer Perceptron 32
Figure 16 Feature Importance Ranking of Random Forest using AUC 33
Figure 17 Feature Importance Ranking of Random Forest using F1 score 34
Figure 18 Frequency Histogram of API Sequence length 35

Figure 19 The frequency distribution of API frequency for adware, Backdoor, Downloader, Dropper, Spyware, ... 35
Figure 20 Frequency Histogram of Idgetprocedureaddress 37
Figure 21 The frequency distribution of Idgetprocedureaddress for adware, Backdoor, Downloader, Dropper, Spyware,Trojan, Virus, Worms 37
Figure 22 Frequency Histogram of NtClose 38
Figure 23 The frequency distribution of ntClose for adware, Backdoor, Downloader, Dropper, Spyware,Trojan, Virus, Worms 38
Figure 24 Frequency Histogram of NtAllocateVirtualMemory 39
Figure 25 The frequency distribution of NtAllocateVirtualMemory for adware, Backdoor, Downloader, Dropper, Spyware,Trojan, Virus, Worms 40
Figure 26 Frequency Histogram of ldrloaddll 41
Figure 27 The frequency distribution of drloaddll for adware, Backdoor, Downloader, Dropper, Spyware,Trojan, Virus, Worms 41

SUMMARY

The exponential growth of malware has created a significant threat in our daily lives, which heavily rely on computers running all kinds of software. Malware writers create malicious software by creating new variants, new innovations, new infections and more obfuscated malware by using techniques such as packing and encrypting techniques. Malicious software classification and detection play an important role and a big challenge for cyber security research. Due to the increasing rate of false alarm, the accurate classification and detection of malware is a big necessity issue to be solved. In this research, eight malware family have been classifying according to their family the research provides four feature selection algorithms to select best feature for multiclass classification problem. Comparing. Then find these algorithms top 100 features are selected to performance evaluations. Five machine learning algorithms is compared to find best models. Then frequency distribution of features are find by feature ranking of best model. At last it is said that frequency distribution of every character of API call sequence can be used to classify malware family.

CHAPTER 1 INTRODUCTION

1.1 Background

In the COVID-19 pandemic the dependency of digital communication multiplies. The internet has become the main channel for effective human interaction and the primary way to work, contact and support each other. During this time, cyberattacks have increased exponentially. Businesses and public-sector organizations are increasingly offering or enforcing “work from home” policies, and social interactions are rapidly becoming confined to video calls, social media posts and chat programs. Many governments are disseminating information via digital means. For example, the UK has made digital the default mode of communication, instructing citizens to rely on official websites for updates to avoid flooding phone-based information services with requests.

In today’s unprecedented context, a cyberattack that deprives organizations or families of access to their devices, data or the internet could be devastating and even deadly. In a worst-case scenario, broad-based cyberattacks could cause widespread infrastructure failures that take entire communities or cities offline, obstructing healthcare providers, public systems and networks. So, it is important to study malware and its functionality.

Today's modern advanced malicious software (malware) has been integrated multi-module systems using sophisticated or obfuscated techniques to attack and target the vulnerable or weak systems. The sophisticated threat actors, Advanced Persistent Threats (APT), continued to make the headlines with audacious politically motivated attacks and thefts on target

organization. Any gaps in network security, software patching, or employee awareness will ruthlessly expose in the wave of destructive malware attacks. A single malicious software can attack and infect thousands or even millions of computers in various ways simultaneously. New malware variants are increasingly becoming in million number as well as viruses, worms, ransomware, adware, spyware, and trojan horses. The danger threats are rapidly increased annually, thus the detection of the malicious program plays an essential role in the cybercrime investigation system. The modern threats could lead to damage to the computer system easily and reduce system processing and performance. Malware analysis and classification are also important in the modern malware detection system to reduce and prevent cybercrimes. Recently, many researchers are interested in analyzing and classifying the variants of malware using the Windows API (Application Programming Interface) call sequences to model malware behavior through static or dynamic analysis. It helps incident responders understand the extent of a malware-based incident and rapidly identify additional systems or hosts that could be affected. The actionable information from malicious software analysis can help an organization more effectively to mitigate vulnerabilities exploited by malware and help prevent additional compromise.

1.2 Motivation and Problem Statement

The detection of modern malware using machine learning methods has become harder due to the complex nature of malware. The information security has become a more important problem to various government, private organizations as well as business because the massive evolution of new malware types lead a major risk to information system. Traditional signatures-based antivirus software may fail to classify unfamiliar suspicious programs to corresponding categories and to identify new variants of malware programs [14].

The anti-virus software uses many obfuscation techniques like packing, metamorphosis and other anti-virtual technologies to evade detection. Thus, some of the existing techniques are developed in the traditional works to detect the threats accurately. However, these techniques lack some major drawbacks such as inaccurate detection, not highly efficient. A new idea is needed to remove these drawbacks of systems. The growth of this new malware can be derived from the known families of malware. Hence, it is very time to give time and effort to classify malware type.

The traditional malware classification and detection systems sometimes might fail to classify and detect the known and unknown malware variants and produce false alarms. They are the main challenges of behavior-based anomaly detection. The vital goal of this research work is to identify the malicious traces reliably, that is, to cut down the False Positive Rate (FPR). To reduce the FPR, the accurate classification of malware is solved by proposing the Malware Feature Extraction Algorithm (MFEA).

The complex program or executables embedded with malicious intentions can infect a numerous amount of computer systems through the Internet. They can be various forms likewise virus, worm, Trojan, bot or rootkit. The occurrence of malware is speedily rising on the Internet and poses a serious threat to computer and information systems. Thus, the battle between malicious code writers and researchers is virtually a never-ending game. Therefore, the scalable and applicable malware classification datasets are provided for malicious classification system.

1.3 Contribution of the research

The main objective of the research is to study the behavior of malware, to analyze the nature and variations of malware, to provide the important features of effective malware family classification, to classify malware families with machine learning, to extract the prominent malicious function calls by using the proposed feature importance, to find out frequency distribution of each malicious function call according to their family.

1, Conduct a literature search on existing malware family classification techniques for static and dynamic analysis with their pros and cons, Study the how malware API works according to family and collect the API call sequence according to malware family.

2. Pre-processing the API call by counting the frequency of API for each sample to feature extract and find out the most important features with comparing four feature

selection techniques such as False discovery rate with ANOVA, RFE, LASSO and FDR with chi-square. Each algorithm selects the first top 100 features.

3. Evaluate the malware family classification by comparing four machine learning algorithms such as SVM (RBF kernel, polynomial and linear), MLP, RF with selected best 100 features.

4. Find out the feature importance based on AUC and F1 score of best classification results.

5. Find out details' frequency distribution of each feature according to the family.

1.4 Organization of Research

The thesis dissertation organized in six-chapter. This chapter describes the introduction, following with a brief introduction along with them the importance of malware analysis on cybersecurity, the problem areas and the solutions of the thesis, the objectives, the contributions, and finally the organization of the thesis.

Chapter 2 discusses the recent research methodologies on malicious threats classification and detection of the cybersecurity field. The evolution of malware and different types of features for static and dynamic analysis have been described in Chapter 2 along with recent research works. Then, different feature selection approaches, and classification methods have been discussed in Chapter 2.

Chapter 3 supports the proposed system of family classification for malicious threats. This chapter highlights the malicious feature extraction algorithms for API sequences. Four

different feature selection algorithms are used. The preprocessing phases have been conducted for malware family classification dataset download from IEEE Data Port.

Chapter 4 highlights the implementation of the malware classification system for malware family. The experiment results are obtained by comparing the result of four machine learning algorithms in terms of ROC-AUC curves.

Chapter 5 describes the feature importance of best results for machine learning algorithms in terms of AUC Curve and F1 score and provides frequency distribution of each feature according to its family.

Chapter 6 concludes with the future work of the research.

Chapter 2 Literature Review

2.1 Overview

In this chapter, we will introduce the previous work of malware classification and feature selection method.

2.2 Introduction

Malware is one of the most common attacks or threats to victims' hosts such as operating systems, network systems, and IoT devices, which can lead to huge damage and disruption on government and non-government organizations. At the beginning of the creating malware it is harmless to both user and computer. Malware writers do not use any encryption method to protect it. Now the things have been changed drastically. Now malware author is paying attention to gain financial profits. They introduce various hiding techniques to exploit vulnerabilities and attacks to the user. According to the report the rate of malware infections has been rising for the last ten years.

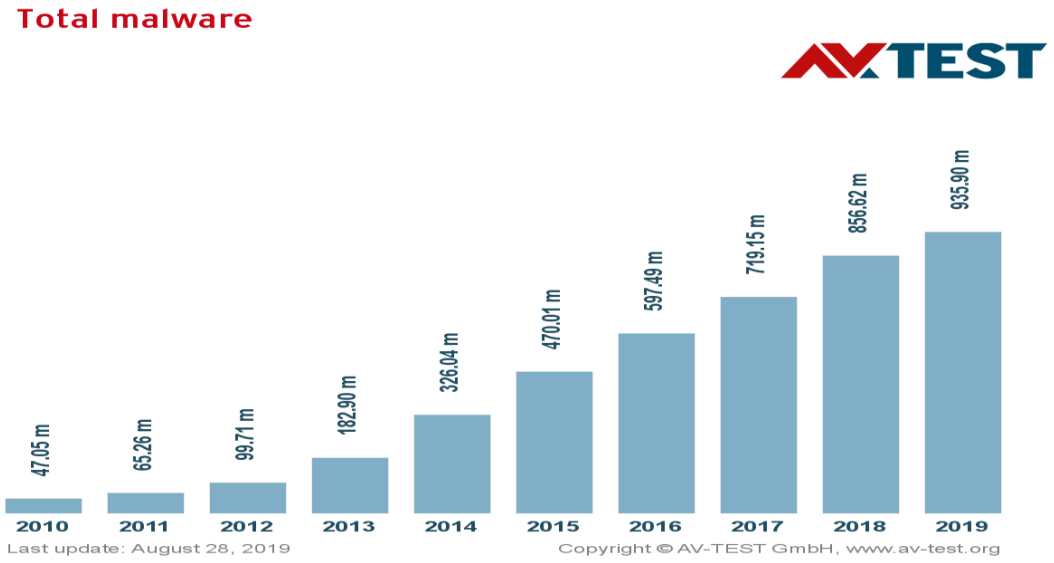

Figure 1.1 The growth of malwares in past 10 years

2.3 Malware Classification

Any software that happens to pose a threat to users, computers, or network systems can be considered as malicious software. It can be various such as scareware, adware, viruses, rootkits, trojan, worms, and spyware, etc. [31]. The following subsection describes common different types of malicious software that can harm the computer system, network security, IoT devices, etc.

Spyware: Spyware refers to programs that use your Internet connection to send information from personal computers to some other computer, normally without knowledge or permission of the owner. This information is a record of the owner's ongoing browsing habits, downloading history or it could be more personal data like username and address.

Downloader: It is a type of trojan that installs itself to the system and waits until an internet connection becomes available to connect to a remote server or website to download additional programs (usually malware) onto an infected computer.

Trojan: A Trojan or Trojan horse is a type of malware that conceals its true content to fool a user into thinking it's a harmless file. The "payload" carried by a Trojan is unknown to the user, it can act as a delivery vehicle for a variety of threats.

Worms: A worm is a type of malware that operates as a self-contained application and can transfer and copy itself from computer to computer.

Adware: Adware is unwanted software designed to throw advertisements up on a computer or device screen, most often within a web browser. Some security professionals view it as the forerunner of the modern-day PUP (potentially unwanted program). Typically, it uses an underhanded method to either disguise itself as legitimate or piggyback another program to trick you into installing it into PC, tablet or mobile device.

Dropper: A dropper is a malicious software that has been designed to install some sort of malware (virus, backdoor, etc.) to a target system. It often carries several completely unrelated pieces of malware that may be different in behavior or even written by different codes. Example: Stuxnet

Virus: A computer virus is a type of malicious code or program written to alter the way a computer operates and is designed to spread from one computer to another. A virus operates by inserting or attaching itself to a legitimate program or document that supports macros to execute its code. In the process, a virus has the potential to cause unexpected or damaging effects, such as harming the system software by corrupting or destroying data. Example: MyDoom

Backdoor: A backdoor refers to any method by which authorized and unauthorized users can get around normal security measures and gain high-level user access (aka root access) on a computer system, network, or software application. Once they're in, cybercriminals can use a backdoor to steal personal and financial data, install additional malware, and hijack devices.

2.3 Feature selection Methods for Malware Classification

The process of selecting the most important attributes is the main role in mining technology. It plays a significant role in improving the classifier's performance and improving the accuracy by discarding or reforming the shape of features or attributes such as the noisy, redundant, and irrelevant from the dataset. To improve the classification rate of malicious and benign executables and to decrease the FP and FN, this stage is essential to find the best API features. Not all the extracted features can be used in training for the following reasons.

- Consuming large memory usage
- Taking long training time for classifiers
- Producing the false alarm rate due to many noisy, redundant or irrelevant features [46].

Feature selection is the most important step for the malware family classification (MFC). As mentioned, one of the objectives is to choose the smallest number of features that keep the classification rate as high as possible to allow to use the minimum quantity of resources for the malware detection task [10]. The efficiency of classification was improved by

implementing the attribute selection technique with the reduced number of attributes for training [9].

2.3.1 Types of Feature Selection

There are three main types of methods in feature selection.

1.Filter method

2.Wrapper method

3.Embedded method

1.Filter Method: In this method features are selected based on their score in various statistical tests for their correlation with the outcome variable. The selection of features is independent of any machine learning algorithms.

2.Wrapper Method:

The feature selection process is based on a specific machine learning algorithm that we are trying to fit on a given dataset in the Wrapper method. It follows a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion. The evaluation criterion is simply the performance measure which depends on the type of problem, e.g. for regression evaluation criterion can be p-values, R-squared, Adjusted R-squared, similarly for classification the evaluation criterion can be accuracy,

precision, recall, f1-score, etc. Finally, it selects the combination of features that gives the optimal results for the specified machine learning algorithm**.**

3.Embedded Methods

Embedded methods perform feature selection as a part of the model creation process. This generally leads to a happy medium between the two methods of feature selection previously explained, as the selection is done in conjunction with the model tuning process. **Lasso and Ridge regression** are the two most common feature selection methods of this type, and the **Decision tree** also creates a model using different types of feature selection**.**

2.3.2 Previous Work on Feature Selection on Malware Detection

In [11] the authors used memory access patterns to distinguish malicious families. The feature selection process was performed and trained by ML Models. They took 50,000 features by selecting the highest IG rank. CFS in Weka was also performed to find the best 10,000 features. However, the accuracy was 0.845% with RF classifiers in their work.

In [12] the researcher use API calls for TF-IDF feature selection algorithm on 552 malicious and benign dataset and achieve 96.4 percent accuracy. In [13] IG feature selection method used a malware and benign dataset and achieve 95.7 percent accuracy.

The authors in [46] used information gain after extracting the n-grams to choose the top 500 features. They experimented on two different datasets: the first data set contains a collection of 1,435 executables (597 clean ware and 838 malware), and dataset2 contains

2,452 executables, (1,370 clean and 1,082 malware). IG attribute selection method was used in [36, 49, 47] and their accuracies with 98%, 94.6, and 97.7% respectively. The accuracy of the hybrid model was 97.4 for both datasets n = 6,4 respectively in [13]. Tf-idf was applied in [15] to get the most relevant features. Among the three FS methods such as filter, wrapper, and embedded methods, most researchers commonly used the filter-based approach in malicious classification and detection research areas. In our work, we first implement three different feature selection techniques and select the best feature set by comparing the result.

2.4 Malware classification in Machine learning

Machine learning has been widely used in the cybersecurity domain. Machine learning has been deployed in many fields such as speech recognition, computer vision, robot control, natural language processing and other applications. Machine Learning has a powerful ability and capability to do many things for cybersecurity. It can be used to identify the APTs which are more complex than the normal malware or threats [16]. Various Machine Learning techniques have been successfully applied to highlights the wide-ranging problems in computer and information security. Machine Learning techniques can be used in many intrusion detection systems (IDS) because it can detect new and unknown attacks.

This section provides some of the related works on malware detection and classification using ML approaches.

Paper 4 The Author compares six different feature selection methods to the exact best feature on n-gram based static analysis of malware sample with 100 samples. For examining the efficiency of the best feature to the antivirus system four different machine

learning algorithms such as SVM, PCA, j48, Naïve Bayes are used. The result shows that the use of Principal Component Analysis (PCA) feature selection and Support Vector Machines (SVM) classification gives the best classification accuracy using a minimum number of features.

In this paper [2] author does a comparative study of several feature selection methods (correlation-based feature selection (CFSSubset) with four different machine learning classifiers in n-gram based static malware detection.

The researcher extracts an API call pattern from 787 malware samples of nine families from malware API call sequences. In this process RNN (Recurrent Neural Network) is used. It gives 71 percent accuracy on average [3].

In paper 5 researchers extract features with four types of feature selection algorithms such as 1-gram Selection 2. 1-gram Paring 3. n-gram Selection 4. n-gram Paring then apply deep feed-forward neural network and convolutional neural network then combining this two neural with ensemble classifier. The using n-gram features trained to be 96.7% accurate. The convolutional neural network, on the other hand, trained to 88.5% accuracy on the test data. By multiplying the nodes by these accuracies, we created weighted nodes. Combining these nodes as an ensemble classifier returned an accuracy of 97.7%. The training time for the feed-forward neural network was much longer than the training for the convolutional neural network. This was expected since the feed-forward neural network was training weights for every input node rather than the training the smaller convolutions.

In paper 7, the Author proposes the training features for malware family analysis and analyzes the multi-classification performance of ensemble models. For malware

classification machine learning algorithm Random Forest and XGBoost are used to extract features from API and DLL information. For family classification using dimension reduction techniques to convert API and DLL information high to low, The proposed feature selection method provides the advantages of data dimension reduction and fast learning. In performance comparison, the malware detection rate is 93.0% for Random Forest, the accuracy of the malware family dataset is 92.0% for XGBoost.

In paper 6, In 2018, the Author made the first part of our Android malware dataset, CICAndMal2017 [16], publicly available while performing dynamic analyses on real smartphones. In this paper, we provide the second part of the CICAndMal2017 dataset [16] publicly available which includes permissions and intents as static features, and API calls as dynamic features. Besides, we examine these features with our two-layer Android malware analyzer. According to our analyses, we succeeded in achieving 95.3% precision in Static-Based Malware Binary Classification at the first layer, 83.3% precision in Dynamic-Based Malware Category Classification and 59.7% precision in Dynamic-Based Malware Family Classification at the second layer.

In paper 8, In this paper author use memory access patterns to distinguish between 10 malware families and 10 malware types. The author uses n-gram techniques to feature extraction and Information gain feature selection methods to select the best features and convert these best features into bitmap images. Then different machine learning algorithms such as KNN, RF, j48, SVM, NB, ANN are used to evaluate the performance. The accuracy is 0.688 for malware type and malware family classification is 0.8.

CHAPTER 3 FEATURE ENGINEERING

3.1 Overview

In this chapter the description of the dataset is given. We discuss how data is preprocessing and extract the most efficient feature for machine learning algorithms.

3.2 Description of dataset

The IEEE dataset is collected from IEEE Data Port. The dataset contains eight main malware families: Trojan, Backdoor, Downloader, Worms, Spyware, Adware, Dropper, Virus. The dataset is constructed by obtaining the MD5 hash values of the malware collected from GitHub where the families are selected from the report of 67 different antivirus software in virus total. The total malware samples are 7059.

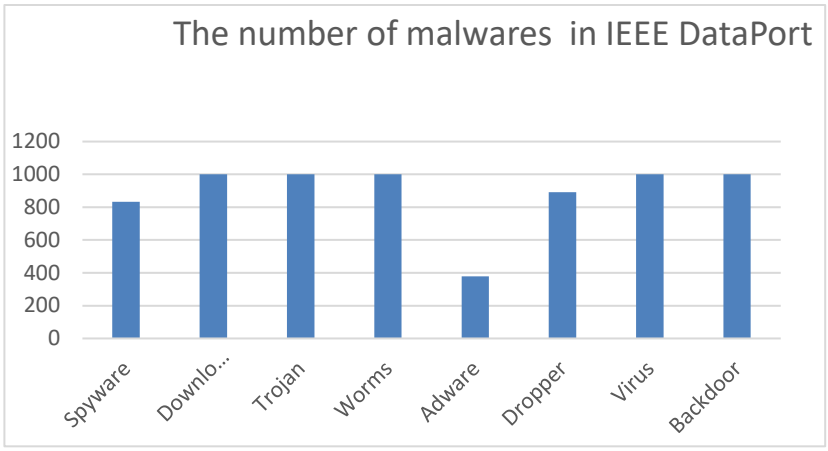

Figure 2 The growth of malware in past 10 years

3.3 Dataset preprocessing

In this experiment, The API call frequency is counted for each sample. The number of each API Call is counted as a feature for each sample. Along with API call frequency we calculate the frequency of every character (such as blank space, comma etc.) at each sample. The total number of features is 761 with 7059 samples.

3.3 Feature Selection

For feature selection, four different algorithms such as FDR-ANOVA, FDR-chi2, RFE, LASSO from three different methodologies like Filter, Wrapper and Embedded System are used. For each feature selection SVM with RBF kernel is used as a machine learning algorithm.

3.3.1 No Feature Selection

At first, we implement Kernel SVM with Gaussian as a kernel parameter to project nonlinear separable data lower dimensions to linearly separable data in a higher dimension in such a way that data points belonging to different classes are allocated to different dimensions.

The SVM with RBF kernel machine learning model is applied to the whole dataset. With no feature selection, the ROC curve is 0.76.

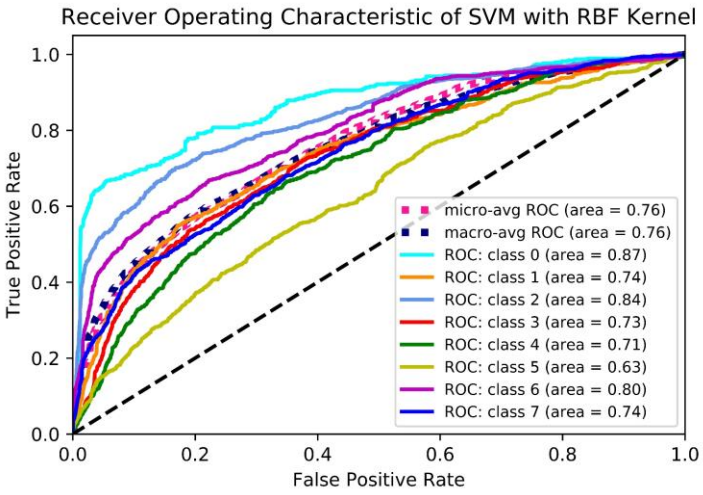

Figure 3 ROC curve for SVM with RBF kernel using no feature selection

3.3.2 FDR Feature Selection

Univariate feature selection was performed by calculating p-values for each feature using analysis of variance (ANOVA) and applying the Benjamin-Hochberg procedure (alpha = 0.005) for multiple comparisons. A univariate statistics-based feature selection method rather than manual selection automates the feature selection process, making it possible to treat the data blindly without assumptions. The ROC score for the top 100 features is 0.81.

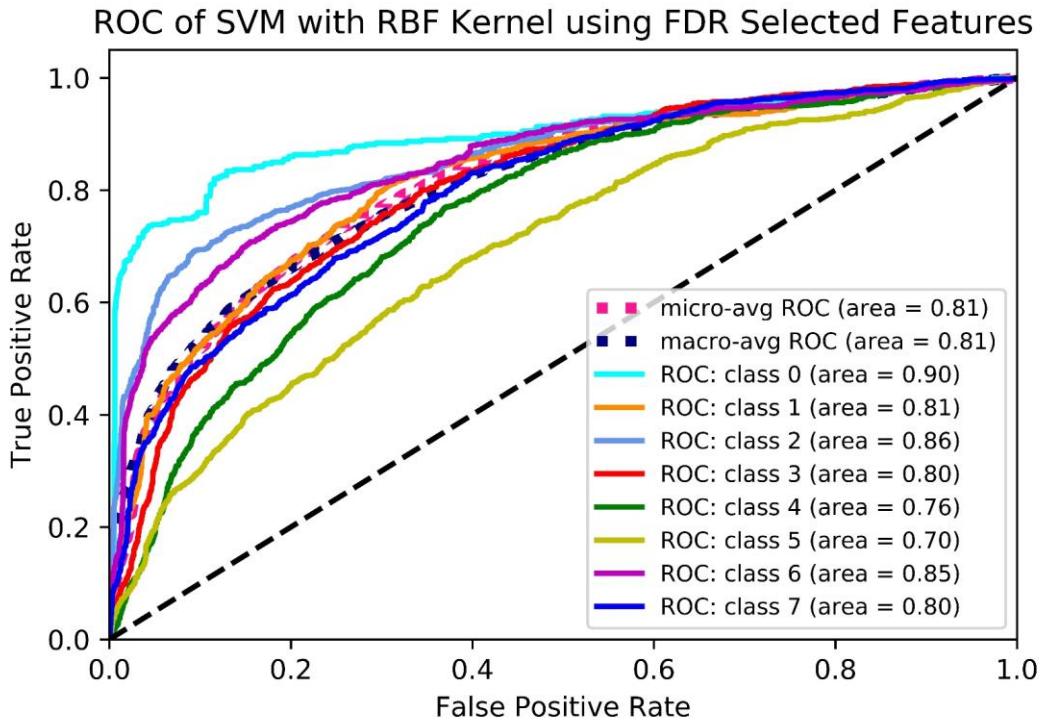

Figure 4 ROC curve for SVM with RBF kernel using 191 features

Then most important 100 features are selected for classification.

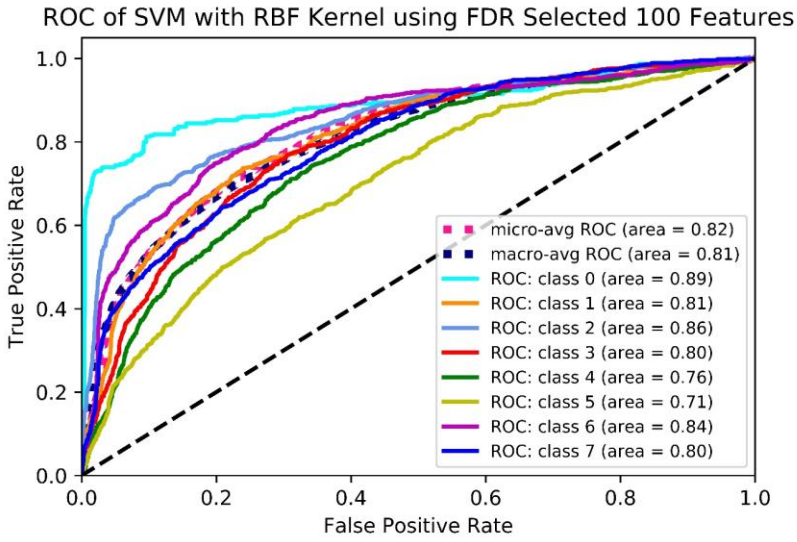

Figure 5 ROC curve for SVM with RBF kernel using 100 features

3.3.3 Chi2-FDR Feature Selection

For each feature we are calculating p values by using the Chi2-Square test. The ROC score for the top 100 features is 0.69.

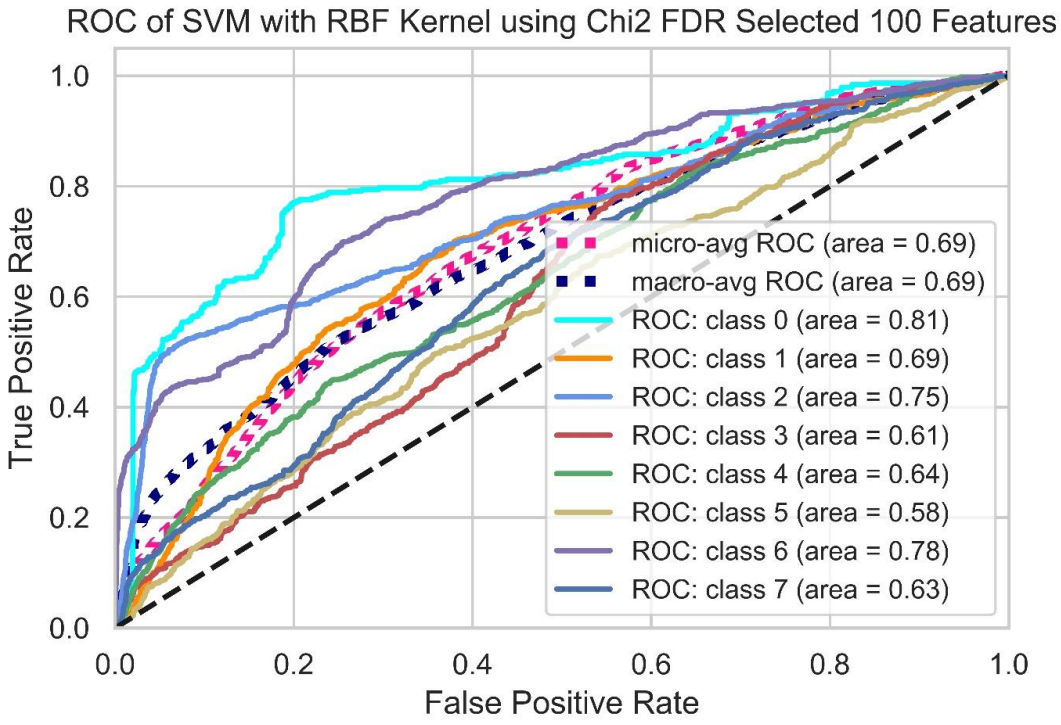

Figure 6 ROC curve of SVM with RBF kernel using Chi2 FDR selected top 100 features

3.3.4 RFE Feature Selection

The recursive feature elimination is to select features by recursively constructing smaller sets of features by evaluating each feature importance with SVM with RBF kernel. The ROC score for the top 100 features is 0.80.

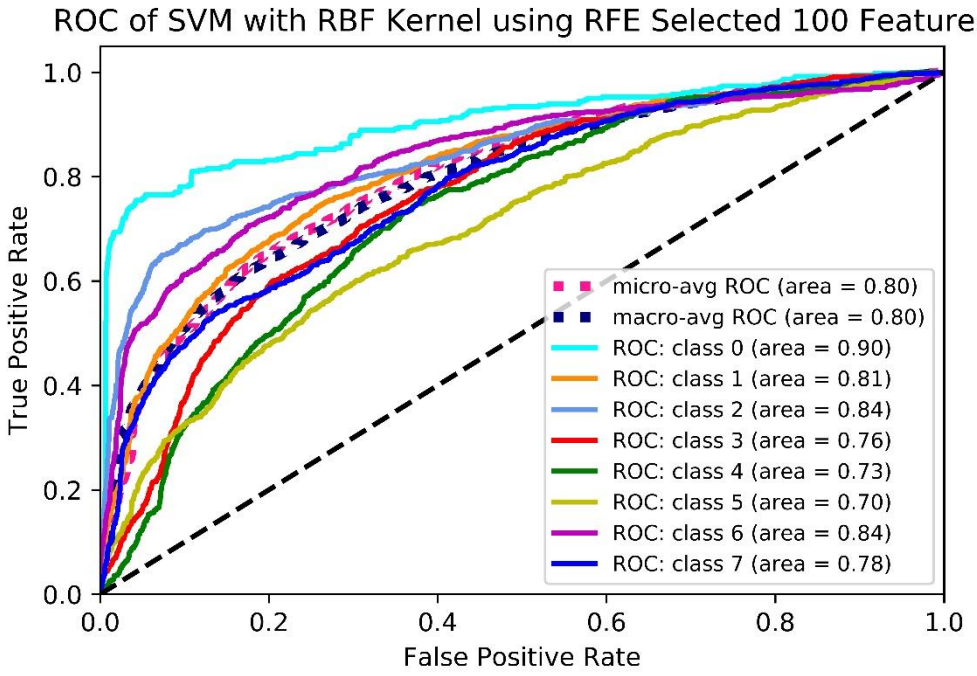

Figure 7 ROC curve of SVM with RBF kernel using RFE Selected 100 Features

3.3.5 Lasso Regression:

The **LASSO method** puts a constraint on the sum of the absolute values of the model parameters, the sum must be less than a fixed value (upper bound). To do so the **method** applies a shrinking (regularization) process where it penalizes the coefficients of the regression variables shrinking some of them to zero. The ROC score for the top 100 features is 0.81.

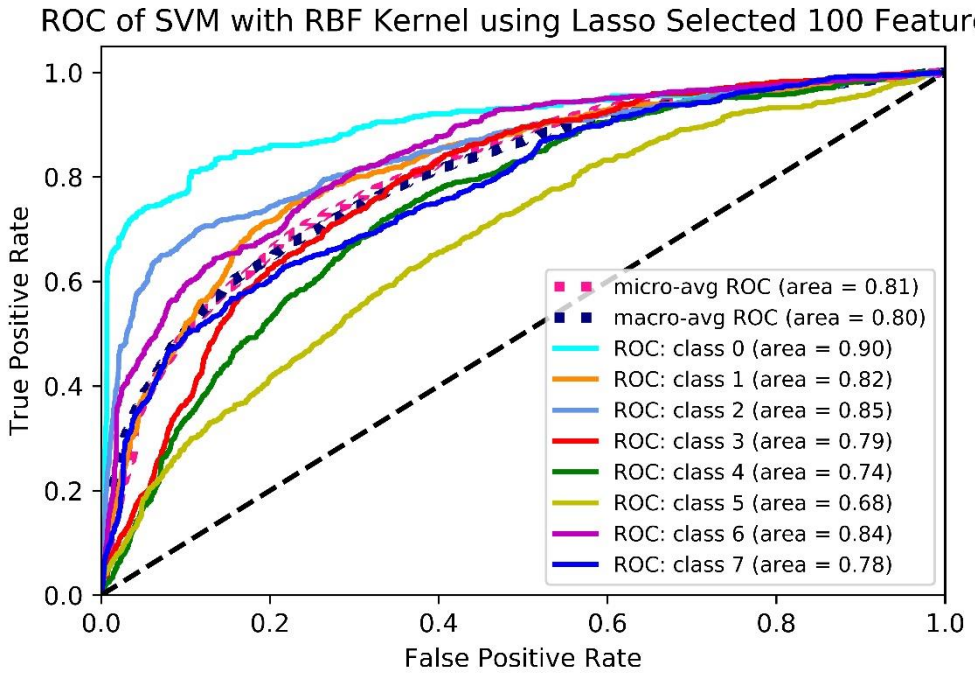

Figure 8 ROC curve of SVM with RBF kernel using LASSO Selected 100 Features

\

3.4 Comparing Results of all Different Feature Selection Algorithms

In this experiment, four different feature selection algorithms are implemented in the malware API dataset for selecting the top 100 features, The FDR with ANOVA gives the best results. The AUC score is 0.82. In the next step different machine learning algorithms are implemented the most important features.

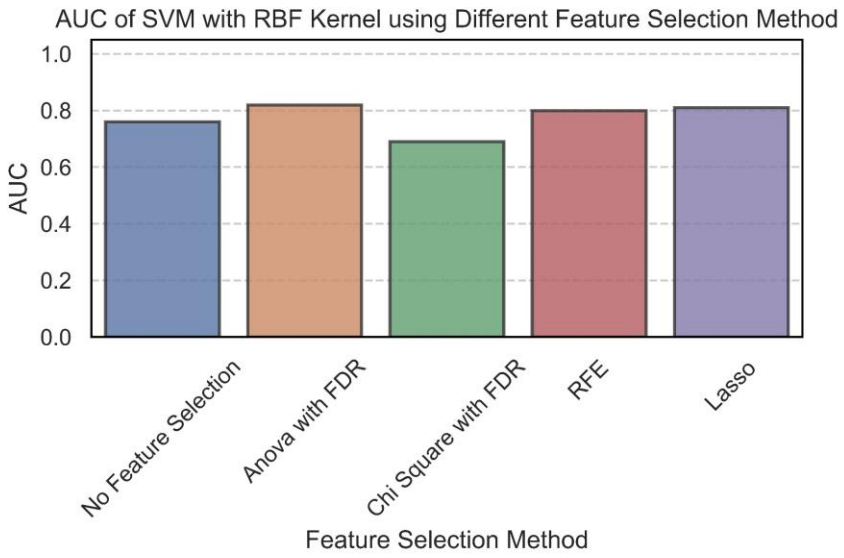

Figure 9 AUC of SVM with RBF kernel using ANOVA,ANOVA with FDR, Chi Square with FDR, RFE, Lasso Feature Selection Method with no feature selection

CHAPTER 4 MACHINE LEARNING

4.1 Overview

In this chapter, four machine learning algorithms are used on 100 features of ANOVA - FDR feature selection algorithm with 10-fold cross-validation and obtain ROC Curve.

4.2 SVM

4.2.1 Linear Kernel

SVM obtains an optimal decision boundary function with a linear kernel that divides the data in such a way that the misclassification error can be minimized q2. For selected top 100 features the ROC score is 0.78.

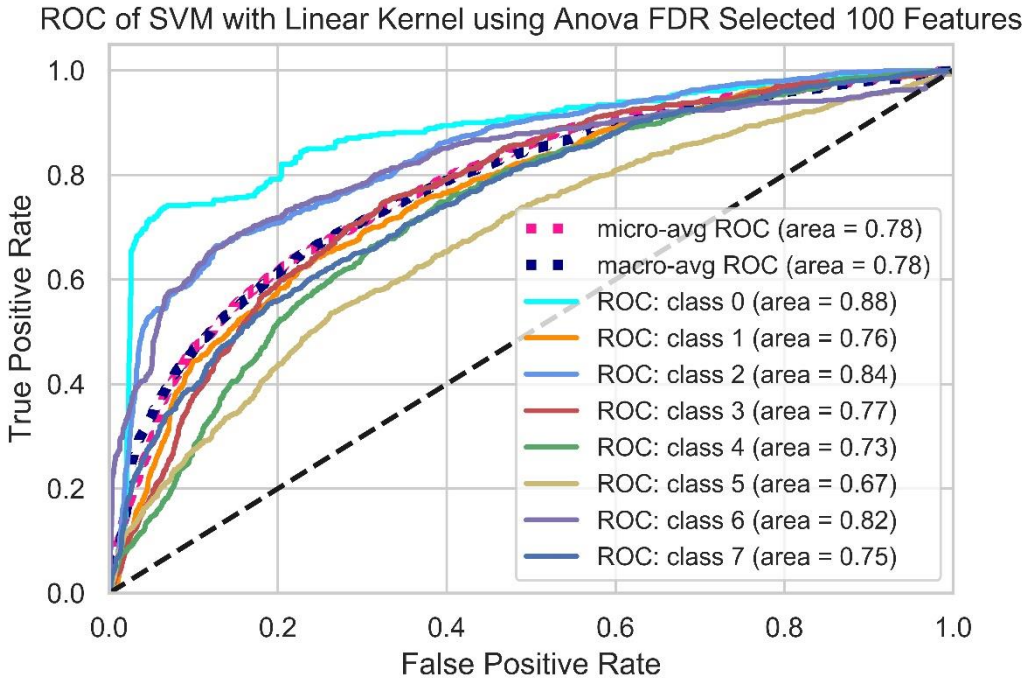

Figure 10 ROC curve of SVM with Linear Kernel using Anova FDR Selected 100 Features

4.2.2 Poly Kernel

SVM obtains an optimal decision boundary function with the poly kernel by controlling the gamma parameter by auto that divides the data in such a way that the misclassification error can be minimized. For selected top 100 features the ROC score is 0.76.

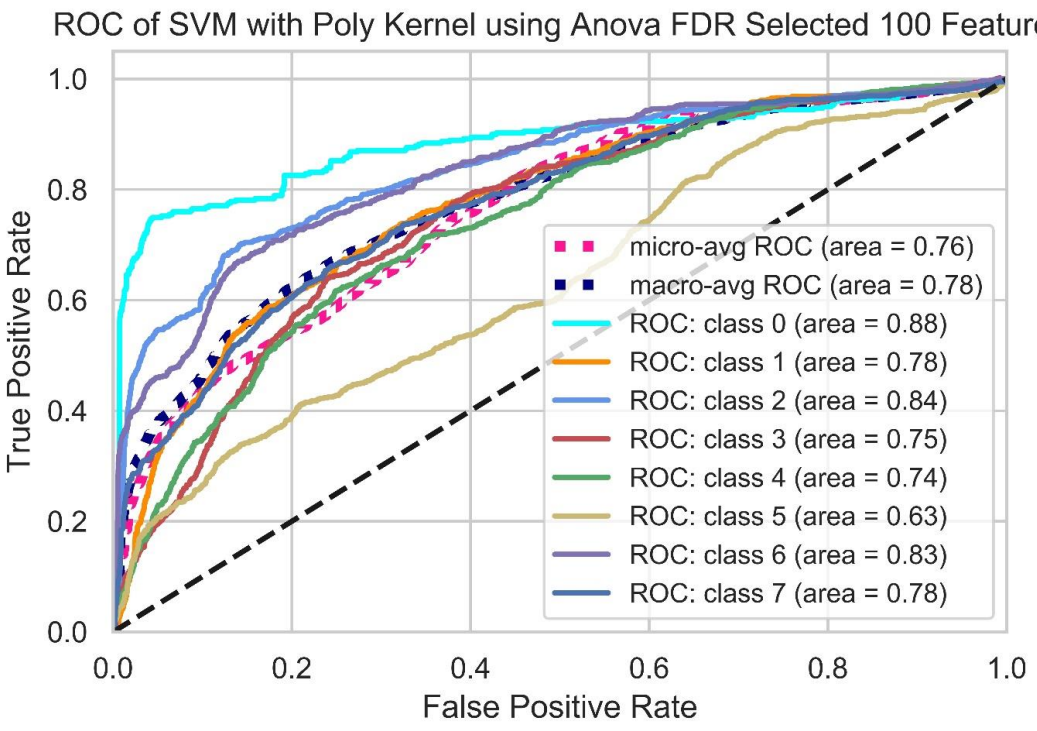

Figure 11 ROC of SVM with Poly Kernel using Anova FDR Selected 100 Features

4.2.3 RBF Kernel

SVM obtains an optimal decision boundary function with RBF kernel by controlling the gamma parameter by auto that divides the data in such a way that the misclassification error can be minimized. For selected top 100 features the ROC score is 0.82

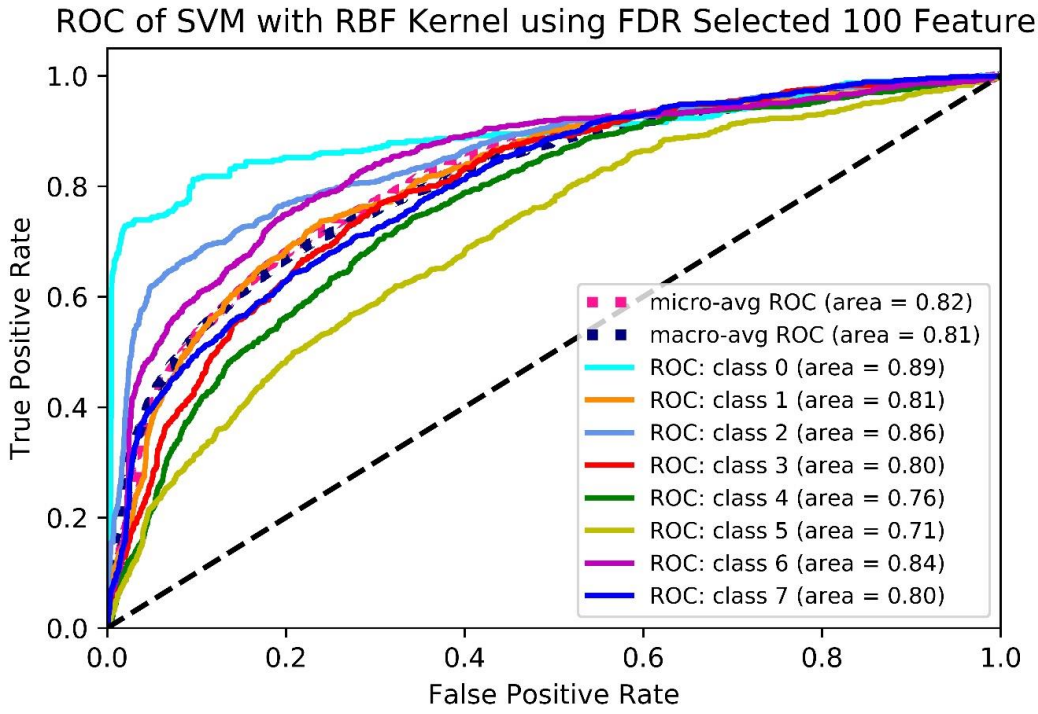

Figure 12 ROC curve of SVM with RBF kernel using FDR Selected 100 Features

4.3 Random Forest

The random forest predicts ROC Curve with 1000 number of decision tree on various sample of dataset and uses averaging to improve the predictive accuracy and controlling over-fitting by max depth 20. For selected top 100 features the ROC score is 0.88.

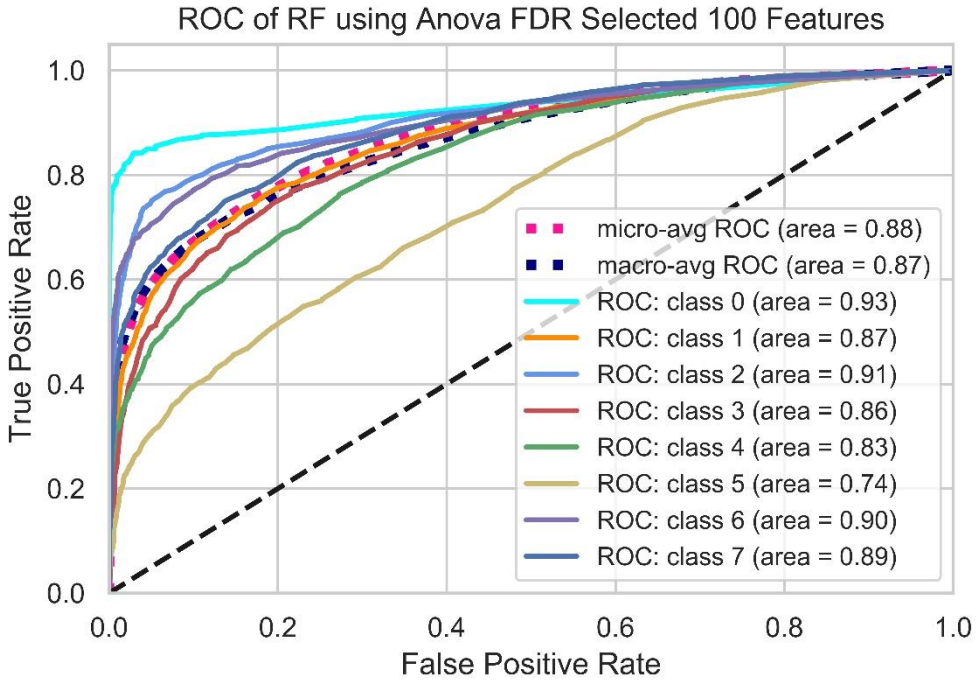

Figure 13 ROC Curve of Random Forest using Anova FDR for top 100 features

4.4 MLP

A multilayer perceptron is a neural network model. In this experiment we use a maximum iteration of 100000 with two hidden layers for training the features and use stochastic gradient descent to optimized log loss function. For selected top 100 features the ROC score is 0.78.

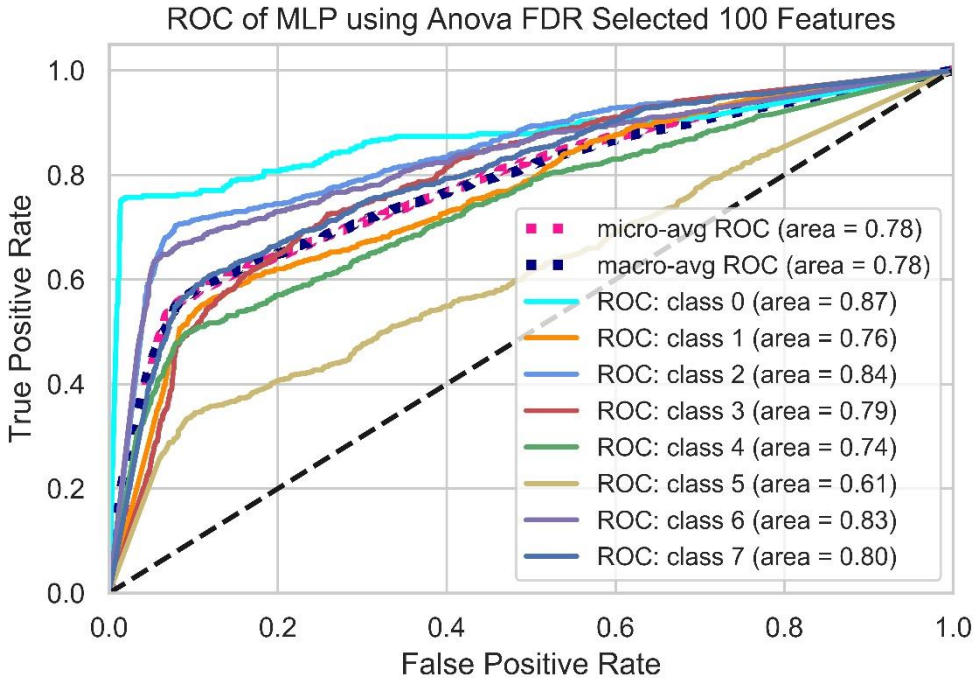

Figure 14 ROC Curve of MLP Anova FDR Selected 100 Features

4.5 Comparing AUC performance for different Classifier

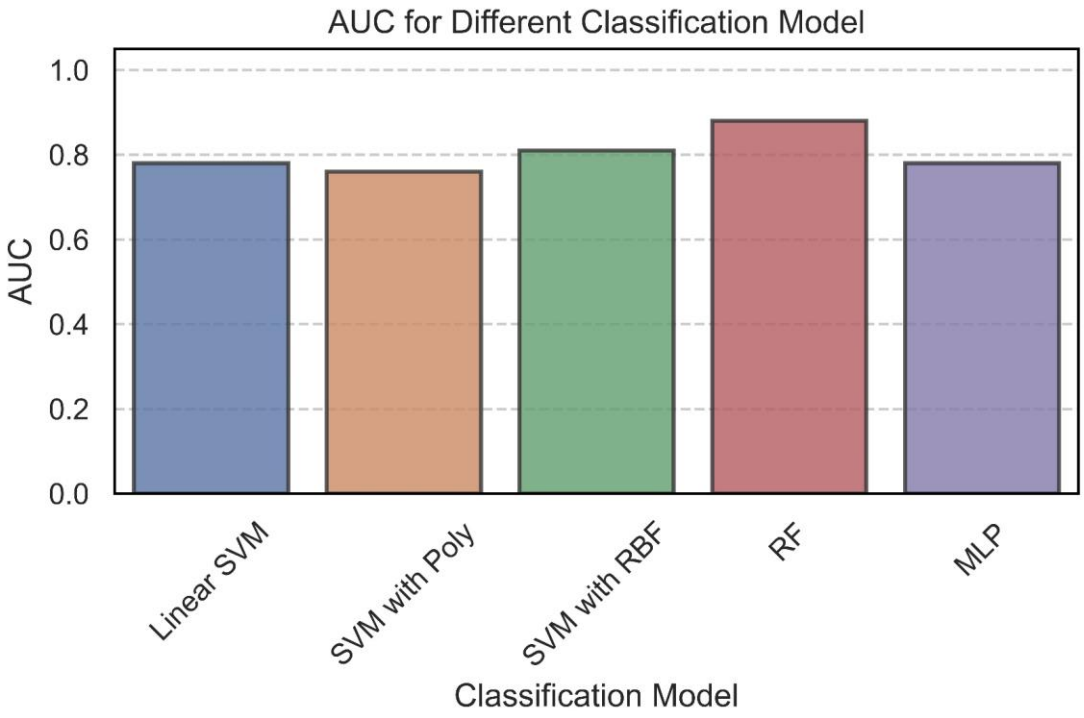

Figure 15 Comparison of AUC Curve in Linear SVM,SVM with Polynomial, SVM with RBF, Random Forest and Multilayer Perceptron

CHAPTER 5 FEATURE IMPORTANCE

5.1 Overview

In this chapter, the most important API call is finding out by obtaining the feature importance of the Random forest classifier model and also given details frequency distribution of each malware feature according to their families.

5.2 Feature importance Ranking using Random Forest Classifier

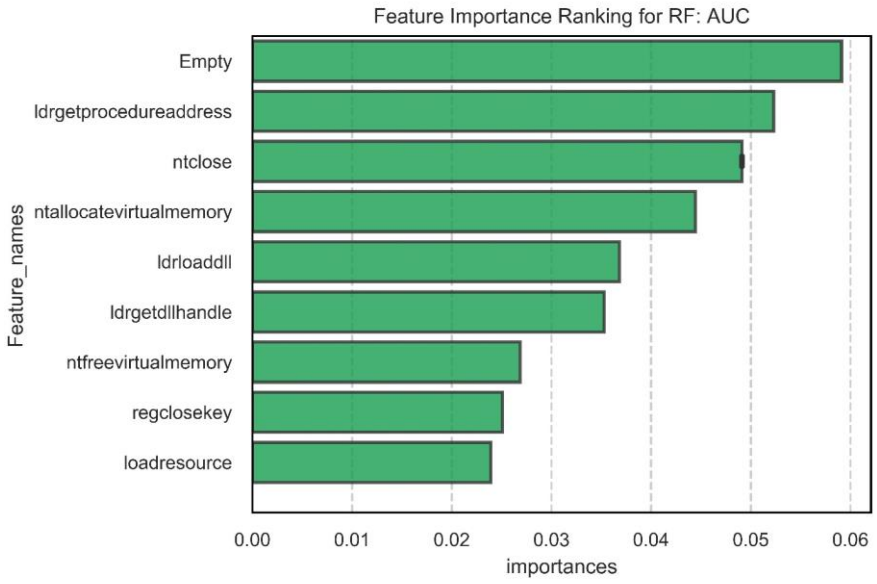

Figure 16 Feature Importance Ranking of Random Forest using AUC

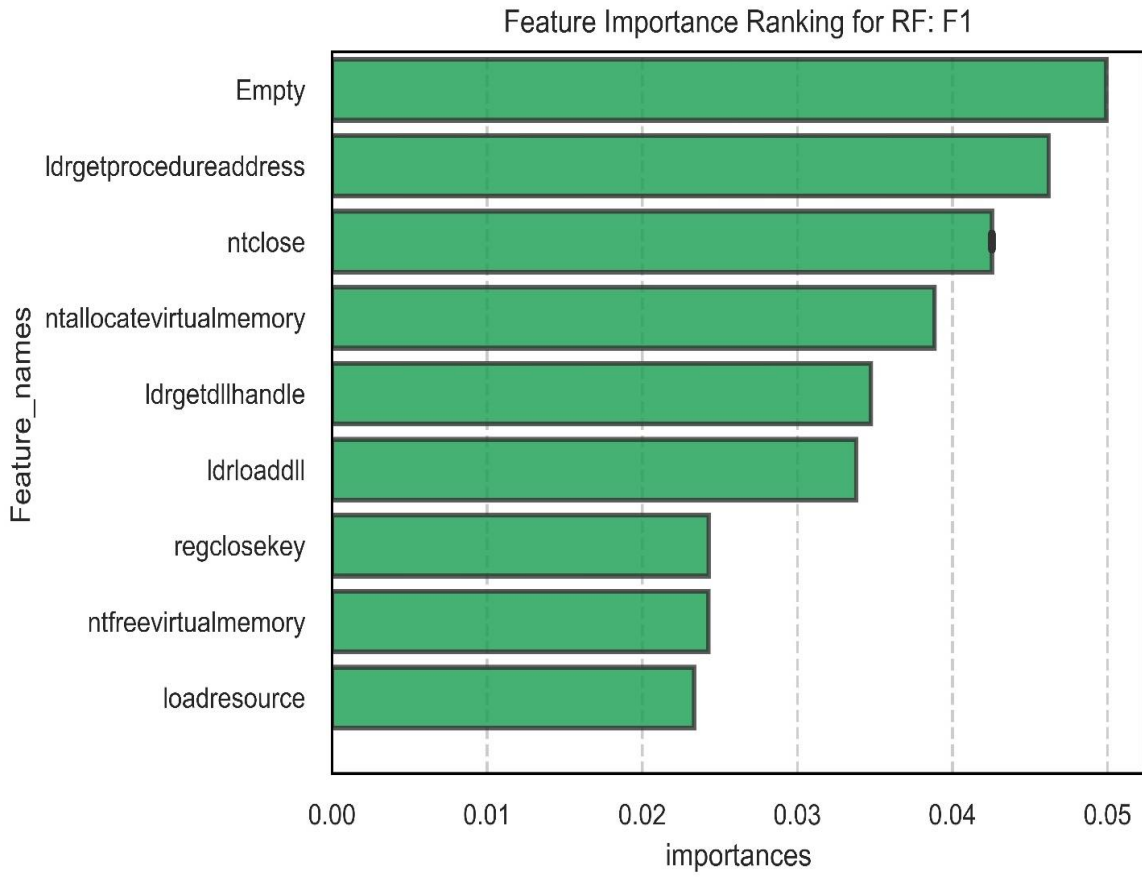

Figure 17 Feature Importance Ranking of Random Forest using F1 score

5.3 Frequency Distribution of Top Feature

5.3.1 The API Sequence length

The API sequence length has the most significant impact on Malware family Classification. We can predict the family of malware by finding the frequency pattern of each malware family.

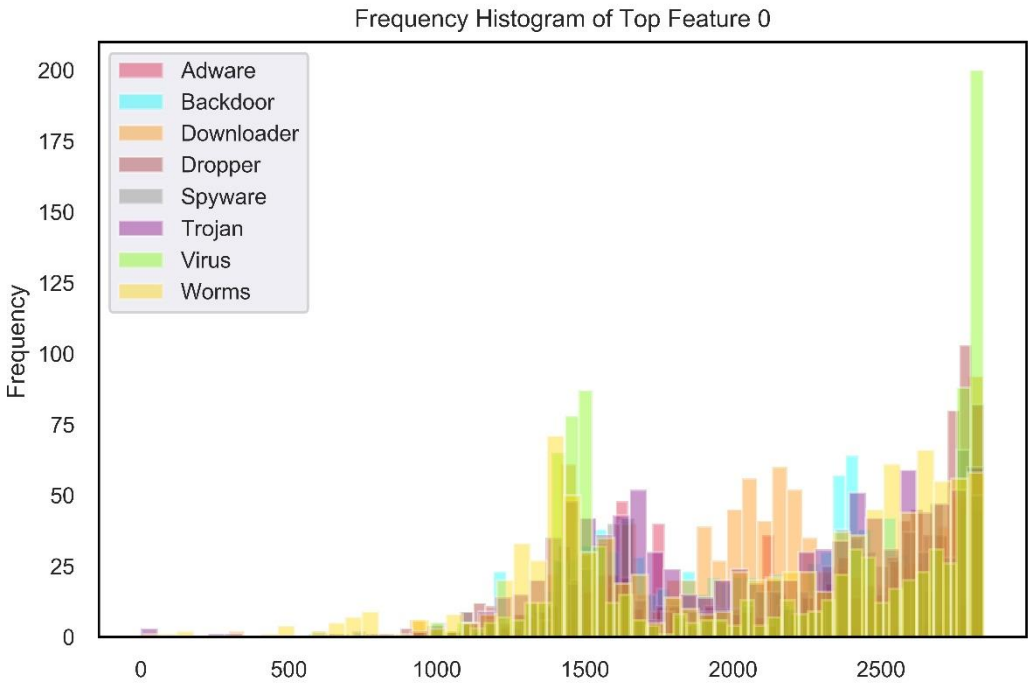

Figure 18 Frequency Histogram of API Sequence length

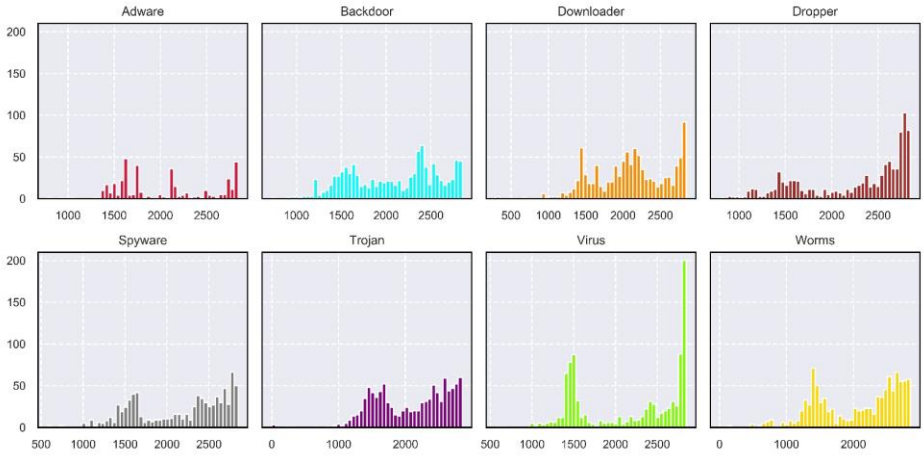

Figure 19 The frequency distribution of API frequency for adware, Backdoor, Downloader, Dropper, Spyware, Trojan, Virus, Worms

5.3.2 Idgetprocedureaddress

The next important API is Idgetprocedureaddress which returns a pointer to the location in memory of the symbol ProcName or Ordinary value Ordinal in the dynamically loaded library.

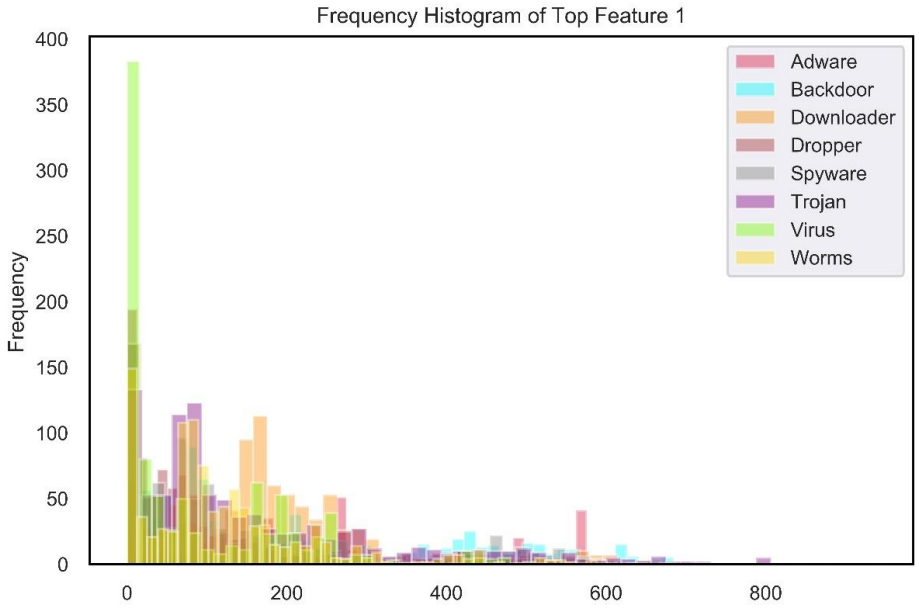

Figure 20 Frequency Histogram of Idgetprocedureaddress

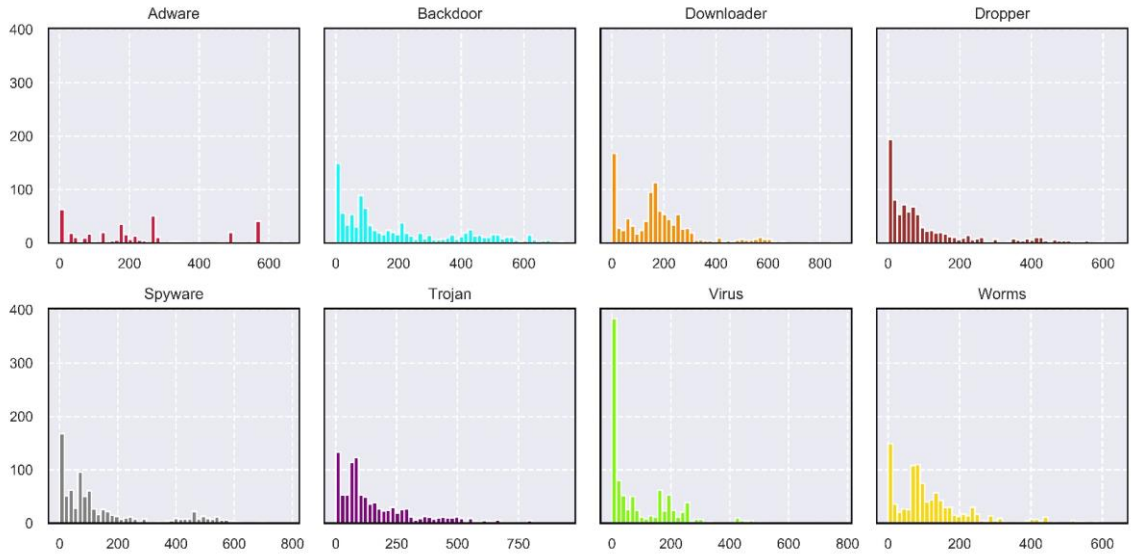

Figure 21 The frequency distribution of Idgetprocedureaddress for adware, Backdoor, Downloader, Dropper, Spyware,Trojan, Virus, Worms

5.3.3 ntclose

The **NtClose** function closes Access token, Communications device, Console input, Console screen buffet, Event, File, Job, Mail slot, Mutex, Named pipe, Process, Semaphore, Socket, Thread.

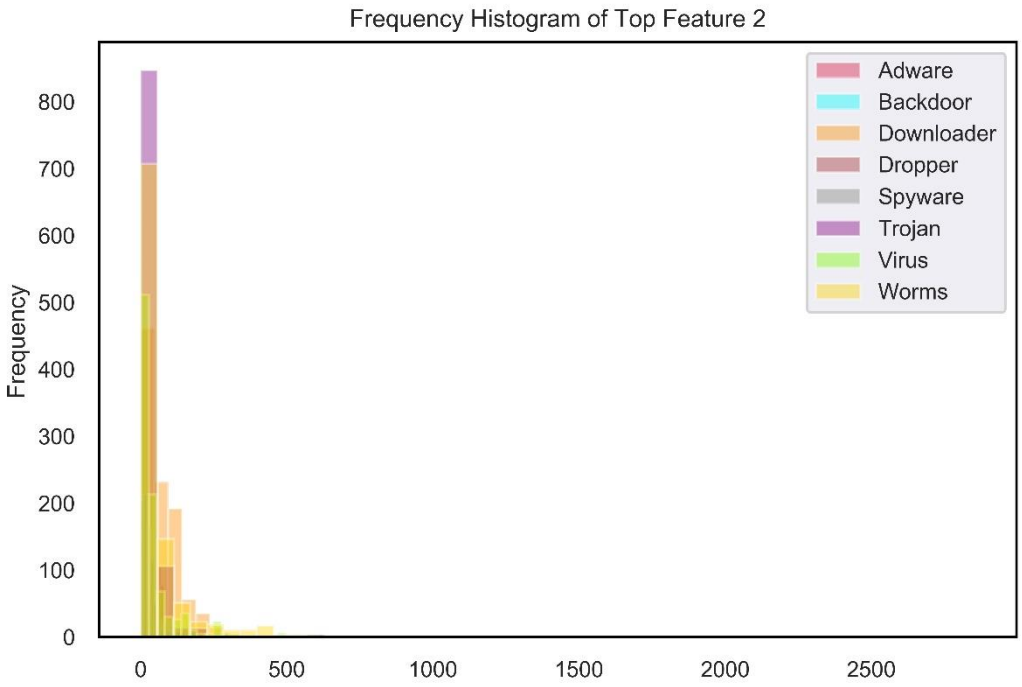

Figure 22 Frequency Histogram of NtClose

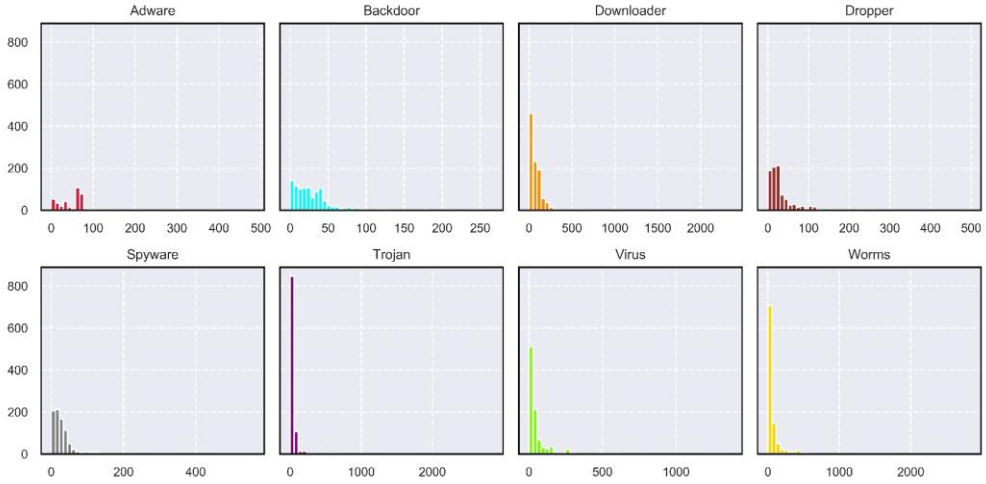

Figure 23 The frequency distribution of ntClose for adware, Backdoor, Downloader, Dropper, Spyware, Trojan, Virus, Worms

5.3.4 ntallocatevirtualmemory

The **NtAllocateVirtualMemory** routine reserves, commits, or both, a region of pages within the user-mode virtual address space of a specified process.

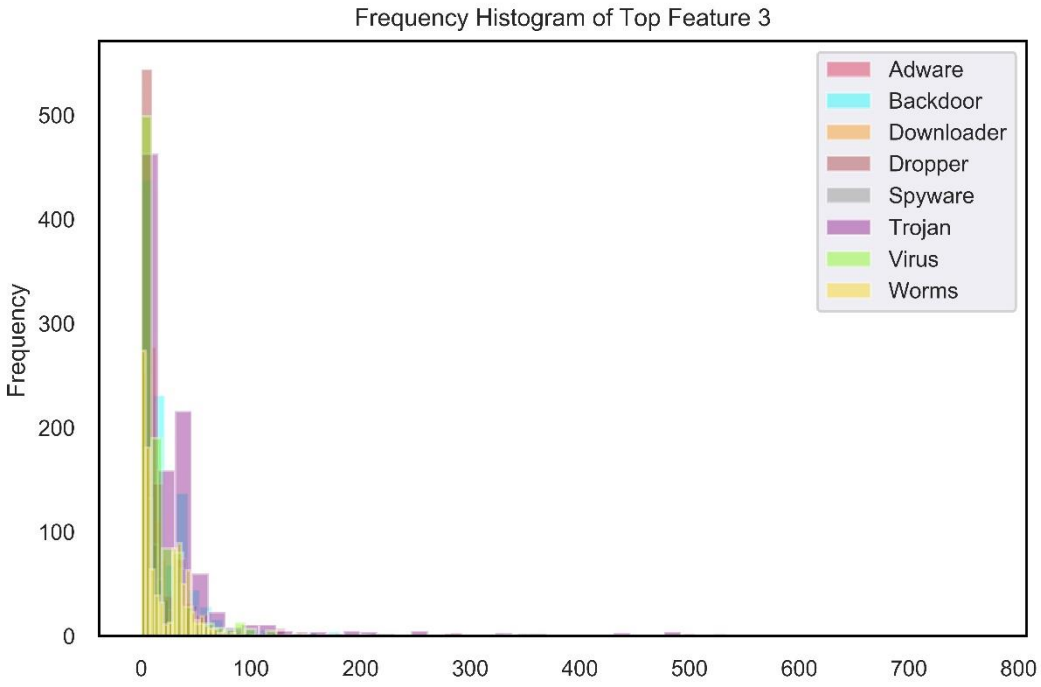

Figure 24 Frequency Histogram of NtAllocateVirtualMemory

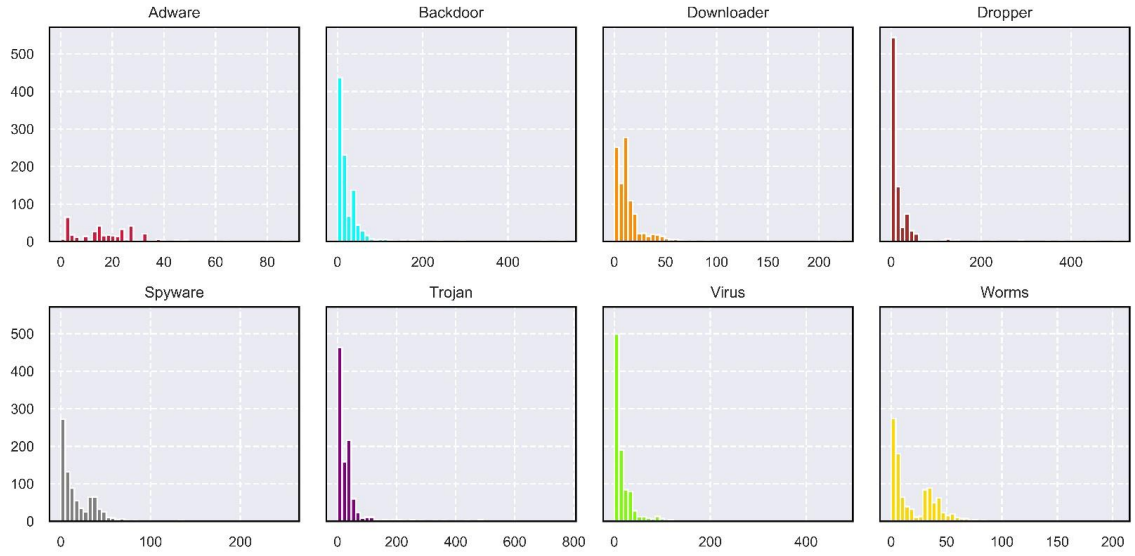

Figure 25 The frequency distribution of NtAllocateVirtualMemory **for adware, Backdoor, Downloader, Dropper, Spyware,Trojan, Virus, Worms**

5.3.5ldrloaddll

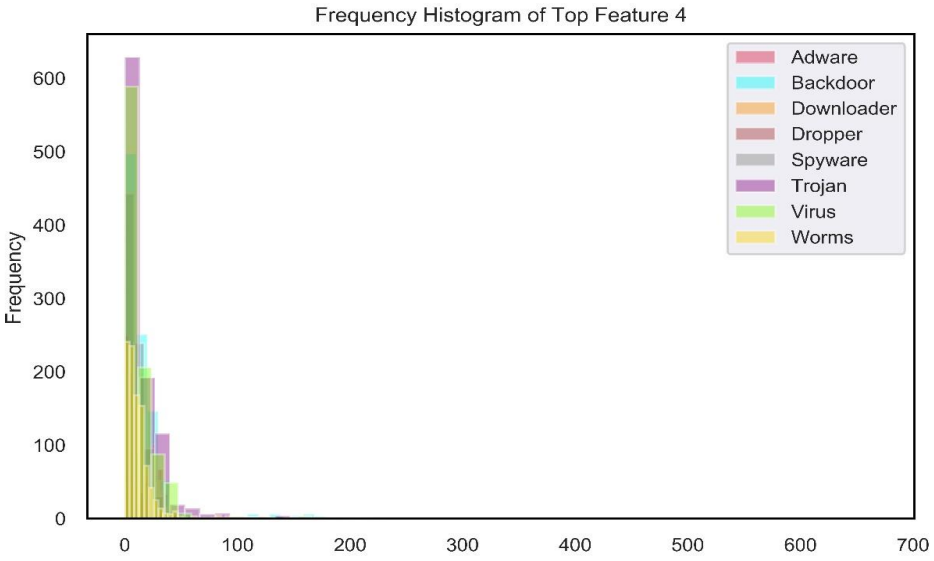

Figure 26 Frequency Histogram of ldrloaddll

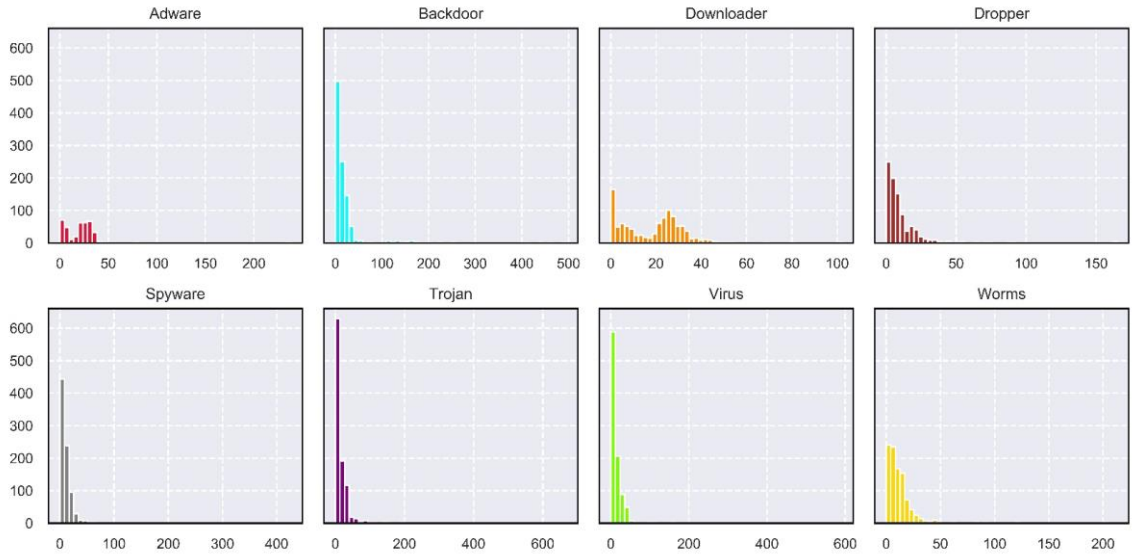

Figure 27 The frequency distribution of drloaddll for adware, Backdoor, Downloader, Dropper, Spyware,Trojan, Virus, Worms

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this experiment, four different feature selection algorithms such as FDR-ANOVA, RFE, LASSO, FDR-Chi2 square are applied on the processed malware family API dataset and figure out the top 100 features by comparing those algorithms with SVM RBF kernel. The best top 100 feature is selected from the FDR-ANOVA feature selection technique where the ROC Curve is 82 percentiles. Then five machine learning algorithms such as SVM algorithms (linear, RBF and polynomial), MLP, Random Forest are implemented on the one hundred most important features from the dataset. By comparing those machine learning algorithms, we find out the best classifier Random Forest. The ROC curve score is 0.82. Then we calculate feature importance concerning AUC and F1 score. Both results are almost the same. We will find API sequence length as the best feature to classify the malware family. Finally, we find out the frequency distribution of the first five best features of malware family classification. In conclusion, we can say that the frequency distribution of each feature including malware API call function can be used as malware family classification.

6.2 Future Work

In the future, for better performance, we increase the number of best features. We use frequency pattern as a feature to classify the malware family. By extending the present work, the unidentified malware and zero-day attack will be developed for the detection system. Deep learning and big data analysis will be kept exploring.

REFERENCES

1. Islam, R., & Altas, I. (2012, October). A comparative study of malware family classification. In *International Conference on Information and Communications Security* (pp. 488-496). Springer, Berlin, Heidelberg.
2. Khammas, B. M., Monemi, A., Bassi, J. S., Ismail, I., Nor, S. M., & Marsono, M. N. (2015). Feature selection and machine learning classification for malware detection. *Jurnal Teknologi*, *77*(1), 234-250.
3. Kwon, I., & Im, E. G. (2017, September). Extracting the representative API call patterns of malware families using recurrent neural network. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems* (pp. 202-207).
4. Mays, M., Drabinsky, N., & Brandle, S. (2017). Feature Selection for Malware Classification. In *MAICS* (pp. 165-170).
5. Taheri, L., Kadir, A. F. A., & Lashkari, A. H. (2019, October). Extensible Android Malware Detection and Family Classification Using Network-Flows and API-Calls. In *2019 International Carnahan Conference on Security Technology (ICCST)* (pp. 1-8). IEEE.
6. San, C. C. (2019). *Effective Malicious Features Extraction and Classification for Incident Handling Systems* (Doctoral dissertation, University of Computer Studies, Yangon).
7. Banin, S., & Dyrkolbotn, G. O. (2018). Multinomial malware classification via low-level features. *Digital Investigation*, *26*, S107-S117.
8. Fan, C. I., Hsiao, H. W., Chou, C. H., & Tseng, Y. F. (2015, July). Malware detection systems based on API log data mining. In *2015 IEEE 39th annual computer software and applications conference* (Vol. 3, pp. 255-260). IEEE.
9. Cepeda, C., Tien, D. L. C., & Ordónez, P. (2016, October). Feature selection and improving classification performance for malware detection. In *2016 IEEE International Conferences*

on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom) (pp. 560-566). IEEE.

10. San, C. C., & Thwin, M. M. S. (2019). Selecting Prominent API Calls and Labeling Malicious Samples for Effective Malware Family Classification. *International Journal of Computer Science and Information Security (IJCSIS)*, *17*(5).
11. Norouzi, M., Souri, A., & Samad Zamini, M. (2016). A data mining classification approach for behavioral malware detection. *Journal of Computer Networks and Communications*, *2016*.
12. Xiaofeng, L., Xiao, Z., Fangshuo, J., Shengwei, Y., & Jing, S. (2018). ASSCA: API based sequence and statistics features combined malware detection architecture. *Procedia Computer Science*, *129*, 248-256.
13. Masud, M. M., Khan, L., & Thuraisingham, B. (2007, June). A hybrid model to detect malicious executables. In *2007 IEEE International Conference on Communications* (pp. 1443-1448). IEEE.
14. Lin, C. T., Wang, N. J., Xiao, H., & Eckert, C. (2015). Feature Selection and Extraction for Malware Classification. *J. Inf. Sci. Eng.*, *31*(3), 965-992.
15. https://blog.malwarebytes.com/threat-analysis/2017/10/analyzing-malware-by-api-calls/
16. https://www.webroot.com/us/en/resources/tips-articles/what-is-trojan-virus
17. https://www.malwarebytes.com/adware/
18. https://encyclopedia.kaspersky.com/glossary/trojan-droppers
19. https://en.wikipedia.org/wiki/Dropper_(malware)
20. https://uk.norton.com/nortonblog/2016/02/the_8_most_famousco.html
21. https://en.wikipedia.org/wiki/Backdoor_(computing)