ANALYSIS OF MACHINE LEARNING ALGORITHMS IN MALWARE DETECTION

A PROJECT REPORT

Submitted by

SINDU GAYATHRI M (715516104046)

ANITH KUMAR S (715516104301)

SIVADASS S T (715516104308)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINNEERING

in

COMPUTER SCIENCE AND ENGINEERING

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH

COIMBATORE 641 062

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2020

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"ANALYSIS OF MACHINE LEARNING ALGORITHMS IN MALWARE DETECTION"** is the bonafide work of **SINDU GAYATHRI M (715516104046), ANITH KUMAR S (715516104301), SIVADASS S T (715516104308)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. R. MANIMEGALAI	**Dr. P. ILANGO**
PROFESSOR	**SUPERVISOR**
HEAD OF THE DEPARTMENT	**PROFESSOR**
Department of Computer	Department of Computer
Science and Engineering	Science and Engineering
PSG Institute of Technology and and Applied Research,	PSG Institute of Technology and Applied Research,
Neelambur,	Neelambur,
Coimbatore-641062.	Coimbatore-641062.

Submitted for the University Project Viva Voice held on _____

INTERNAL EXAMINER	EXTERNAL EXAMINER

ABSTRACT

Nowadays, computers are not efficient without internet. Millions of malware samples are out there in the internet. These malwares can cause harm to our systems and network. It is significant that our system has to be protected from these malwares. There are many ways for protecting the system from malicious code. Recently machine learning and deep learning methods are preferred to detect and prevent the malwares rather than traditional methods. Since these algorithms are more efficient than the traditional methods. There are number of algorithms and machine learning models are available. In this Project, Three algorithms namely logistic regression, Random forest and AdaBoost algorithm. In this project The Performance in detecting malwares among three algorithms is compared. In three algorithms, Random forest algorithm shows the higher amount of accuracy in detecting malwares than Logistic regression and AdaBoost Algorithms.

ACKNOWLEGEMENT

I am very grateful to **Dr.P.V. Mohanram**, **Principal** for providing me with an environment to complete our project successfully.

I also take the privilege of expressing my gratitude to **Dr. G. Chandramohan, Vice Principal** in extending his support to carry out our study.

I deeply indebted to **Dr. R. Manimegalai**, **Head of the Department**, Computer Science and Engineering, who molded us both technically and morally for achieving greater success in life.

We are grateful to **Dr. P. Ilango, Professor**, for being instrumental in completing the project work with his complete guidance.

I express my sincere thanks to **Mrs. Bhuvana, Project Coordinator** for her constant encouragement and support throughout my course.

I extend my gratitude to **Mr. C.P. Shabariram, Project Coordinator** for his constant encouragement and support throughout my course.

Finally, I take this opportunity to extend my deepest appreciation to my family and friends, who supported me during the crucial times of our project.

SINDU GAYATHRI M (715516104046)

ANITH KUMAR S (715516104301)

SIVADASS S T (715516104308)

VeriGuide - Originality Report

Individual Report

Background Information

File Name: Final_Report.docx

Report Generated On: 18 /09/2020, 12:39:07 PM

Similarity Statistics Overview

Similar Sentence(s) Found By VeriGuide: 65 out of 458 sentences = 14.19%

Similar Sentence(s) Filtered by User: 65 out of 458 sentences = 14.19%

Sentence(s) Selected By User To Export: 0

Similarity Statistics for Each Source

Entry	Source	From	Similarity
1	https://arxiv.org/pdf/1707.09425	Internet	23 / 458 = 5.02%
2	https://towardsdatascience.com/randomforest-a-powerful-ensemble-learningalgorithm-2bf132ba639d	Internet	9 / 458 = 1.97%
3	https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html	Internet	8 / 458 = 1.75%
4	https://machinelearningmastery.com/supervise d-and-unsupervised-machine-learningalgorithms/	Internet	6 / 458 = 1.31%

5	https://www.tutorialspoint.com/machine_learni ng_with_python/machine_learning_with_pytho n_classification_algorithms_random_forest.htm	Internet	6 / 458 = 1.31%
6	https://en.wikipedia.org/wiki/Machine_learning	Internet	5 / 458 = 1.09%
7	https://towardsdatascience.com/introductionto-logistic-regression-66248243c148	Internet	4 / 458 = 0.87%
8	https://www.hindawi.com/journals/scn/2020/75 01894 /	Internet	4 / 458 = 0.87%
9	https://www.researchgate.net/publication/3225 13385_A_state-of-the-art_survey_of_malware_detection_approaches _using_data_mining_techniques	Internet	4 / 458 = 0.87%
10	https://en.wikipedia.org/wiki/Malware	Internet	3 / 458 = 0.66%
11	https://towardsdatascience.com/understanding - random-forest-58381e0602d 2	Internet	3 / 458 = 0.66%
12	http://ijarcet.org/wp-content/uploads/VOLUME2- ISSUE-6-2037-2039.pdf	Internet	2 / 458 = 0.44%
13	https://towardsdatascience.com/decision-treealgorithm-explained-83beb6e78ef4	Internet	2 / 458 = 0.44%
14	https://www.analyticsvidhya.com/blog/2016/12 /detailed-solutions-for-skilltest-tree-basedalgorithms/	Internet	2 / 458 = 0.44%
15	http://www.cs.columbia.edu/~jdd/papers/isca1 3 _malware.pdf	Internet	1 / 458 = 0.22%
16	https://www.analyticsvidhya.com/blog/2017/09 /30-questions-test-tree-based-models/	Internet	1 / 458 = 0.22%
17	https://www.kaspersky.com/resourcecenter/definitions/what-is-cyber-security	Internet	1 / 458 = 0.22%
18	https://www.math.ust.hk/~makchen/MSBD501 3 /Ch8slide.pdf	Internet	1 / 458 = 0.22%
19	https://www.oracle.com/technetwork/middlewa re/registry/osr111productdocumentation159992 .pdf	Internet	1 / 458 = 0.22%

20	https://www.vebuso.com/2020/02/linear-tologistic-regression-explained-step-by-step/	Internet	1 / 458 = 0.22%

Similarity Details Selected By User

(No suspected sentences were selected by user to export.)

This report was exported on <2020-09-18 12:44:33>.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbrevation	Expansion
ABM	AdaBoost
API	Application Program Interface
BFGS	Broyden-Fletcher-Goldfarb-Shanno Algorithm
FP	False Positive
RF	Random Forest
TP	True Positive
WEKA	Waikato Environment for Knowledge Analysis

CHAPTER 1

INTRODUCTION

1.1 Introduction

Millions of new malware samples have been discovered every day. These newly invented malwares are more malicious and undermines the effectiveness of the traditional signature-based approach.

Malwares, which is also known as malicious software are nothing but the softwares that cause harm to the computer, server, client or network. Generally malwares are intentionally designed to cause damages.

There are many varieties of malwares. Some of them are computer viruses, worms, Trojan horses, ransomware, spyware, adware and scareware.

1.1.1 Viruses

A computer virus is a code that replicates itself by inserting into other programs. A program in which a virus has inserted itself into is also become infected, and is referred to as the virus's host.

1.1.2 Worms

A computer worm replicates itself by executing its own code independent of any other program. The main difference between a virus and a worm is that a worm does not need a host to cause harm as like viruses. Another difference is that how they propagate themselves. In general, viruses attempt to spread through programs/files on a single computer system.

However, worms spread through network connections with the goal of infecting as many computer systems connected to the network as possible.

1.1.3 Trojan horses

A Trojan horse is a malware embedded by its designer in an application or system. The application or system appears to perform some useful function but is performing some unauthorized action .Trojan horses are typically associated with accessing and sending unauthorized information from its host. Such Trojan horses can be classified as spyware as well.

The damages that can be caused due to malwares include hardware failure, data loss, data theft, blocking the communication path or acquiring the working memory etc.

Detecting malware can be categorized broadly into two categories: anomaly-based detection and signature-based detection. In anomaly based detection the intrusion is classified into normal or anomalous based on the activity of it that is monitored.

Signature-based detection approach identifies the presence of a malware infection or instance by matching at least one byte code pattern of the software in question with the database of signatures of known malicious programs, also known as blacklists.

Since the malwares are evolving rapidly everyday it becomes more difficult for the traditional malware to detect them efficiently. This problem can be efficiently solved by Machine Learning approach.

1.2 General Machine Learning

Machine Learning is a subset of Artificial Intelligence. Data Mining is a field within Machine Learning, and focuses on data analysis. It is used in a wide variety of applications, such as email filtering, malware detection and computer visions. Machine Learning algorithms builds a sample data known as the training data to make predictions without being programmed to perform the task. Machine learning is used daily in all the sectors, domains, etc... Across business domains, machine learning is also called as predictive analysis.

There are two general classes of Machine Learning techniques. They are

1. Supervised Learning
2. Unsupervised Learning

Supervised learning is learning where we train the machine using data which is labeled. In supervised learning, the training data is a series of labeled examples. The algorithm is given features and outputs for a particular dataset and must apply what it learns from this dataset to predict the outputs for another given dataset.

Unsupervised learning consists of examples where the feature set is unlabeled. These algorithms generally cluster the data into distinct groups. Supervised learning can be further broken down into two. They are

1. Classification Problems
2. Regression Problems

A classification problem is where the output variable is categorized into disease and no disease. A regression problem is where the output variable is a real value like dollar or weight.

1.3 Motivation:

Mainly we focus on dynamic features of the following 4 categories for regression: file system category, registry category, API call category, and the category of other miscellaneous features.

1.3.1 API Call Category

API (Application Programming Interface) calls are the functions given by the operating system to allow application programs the access to basic operations such as disk reading and process control. Though these calls may ease the process of manipulating the resources of the machine, also provides hackers with a tonnes of possibilities to obtain confidential information. For this API category, we focus on the amount frequency of invoking of the API calls as a set of features.

1.3.2 Registry Category

The registry is a database that holds the global configuration of operating system in a hierarchical manner. Normal programs frequently use the registry to store information like program location and program settings. Thus it makes the registry system like a gold mine of information for malwares, which may refer to it for information such as the location of the local browsers or the version of the host operating system. We extract the following 4 features for registry category: the number of registries being written, opened, read and deleted.

1.3.3 File System Category

File system is the organized data managed by operating system. It consist of two basic parts: file and directory. File system-related features are a crucial set of

Features to focus on since malware has to deal with the file system in one way or another in order to cause harm to the system or to steal confidential information. We consider the following file-related features: the number of files being opened, in existence, written, moved, deleted, failed, read and copied. Adding to this, we also consider the following 3 directory related features: the number of directories being created, enumerated and removed.

1.3.4 Miscellaneous Category

Adding to out-of-the-box functionalities, Cuckoo sandbox is further improved by a set of signatures contributed by the public community. These signatures can identify certain characteristics of the analysed binary such as the ability to detect virtual environment or execution delay time. All these features are good indicators for the high risk level of an executable but may just be false positives. We take the binary characteristic of whether the community signatures are triggered or not. In addition, we also consider 3 other features that may be relevant to the behaviour characterization: number of processes started, the number of mutex created and the depth of the process tree.

1.4 Approaches:

1.4.1 Logistic regression

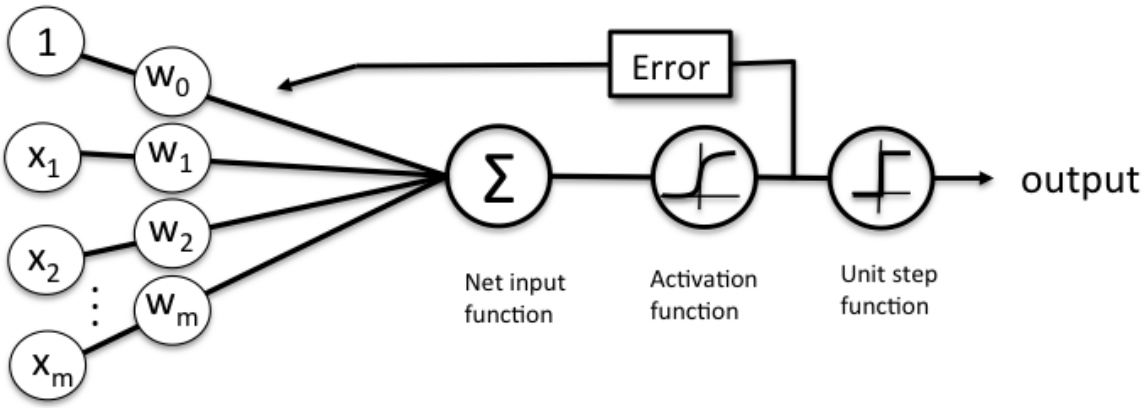


Figure 1 Schematic of a logistic regression classifier

1.4.1.1 Introduction

Logistic regression is basically a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression where the output are continuous values, logistic regression change its output using the sigmoid function to give a probability value which later can be mapped to two or more discrete classes. Figure 1 shows that each input like 1, x1, x2…. Have assigned certain weights w0, w1, w2…respectively this output is given to Net input function which is followed by Activation function. If there is no error it is given to unit step function.

1.4.1.2 Comparison to linear regression

Assume data on time used for studying and exam results. Prediction varies between Linear Regression and logistic regression.

Linear regression predictions are continuous range of values (numbers in a range). It help us predict the student's exam score on a scale of 0 - 100.

Logistic regression predictions are discrete (only specific values or categories are allowed).It help us predict whether the student passed or failed. We can also view probability scores underlying the model's classifications.

1.4.1.3 Types of logistic regression

1. Binary (Pass/Fail)
2. Multi (Horses, Lions, Sheep)
3. Ordinal (Worst, Moderate, Best)

1.4.1.4 Sigmoid activation

The sigmoid function is utilized in order to map the already predicted values to probabilities that are obtained, the function maps any real time value into another value across 0 and 1. We use sigmoid to map predictions to probabilities in machine learning.

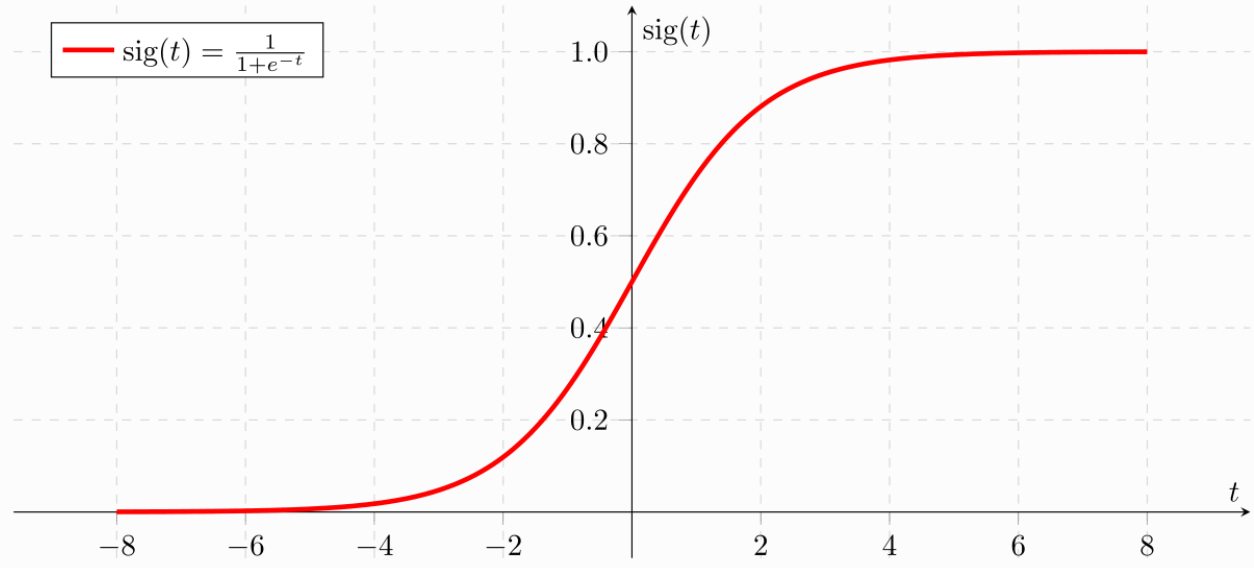


Figure 2 Sigmoid function

1.4.1.5 Decision boundary

The present prediction function returns a probability score between 0 and 1. To map this to a discrete class (correct/incorrect, lion/tiger), a threshold value is selected above which we will classify the values into class 1 and below which we categorize the values into class 2.

$p \geq 0.5$, class=1

$p < 0.5$, class=0

For logistic regression with multiple classes we select the class with the highest prediction probability.

For instance, if our prediction function returned was 0.2 we classify the observation as negative. If the threshold value is 0.5 and our prediction function returned 0.7, we would classify this observation as positive.

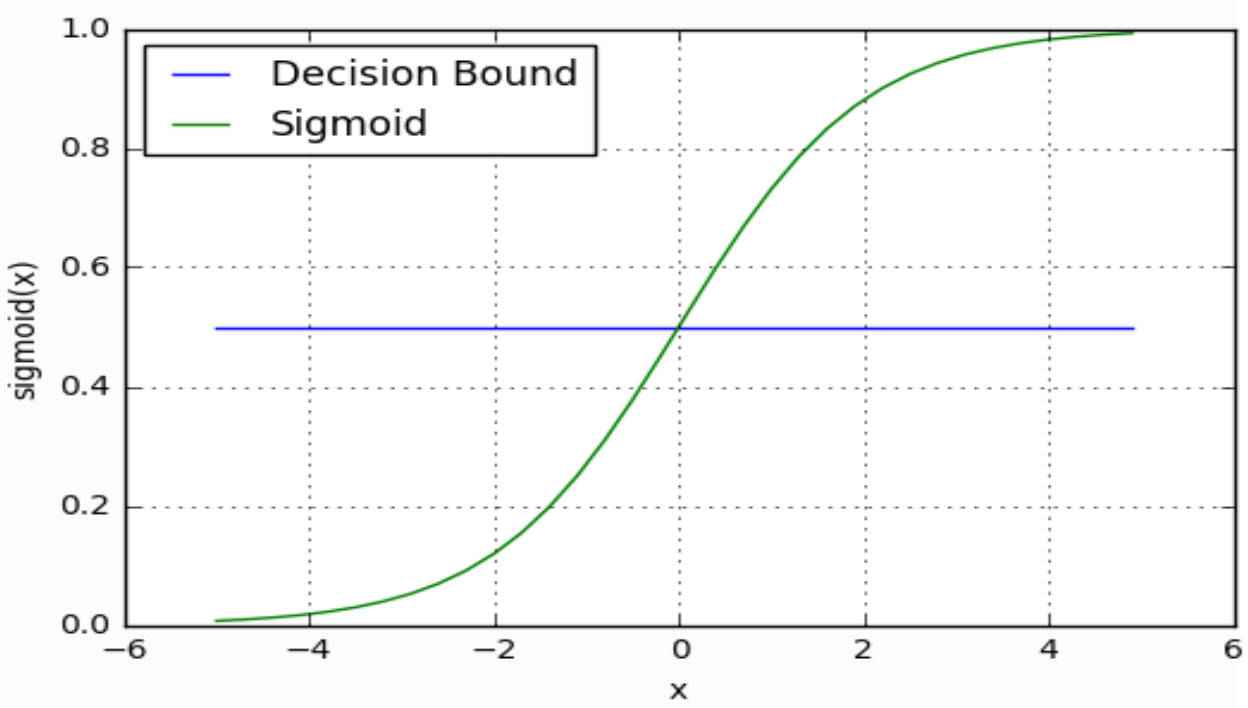


Figure 3 Decision Boundary

1.4.1.6 Cost Function

$$\text{Cost}(h_\Theta(x), Y(\text{actual})) = -\log(h_\Theta(x)) \text{ if } y=1$$
$$-\log(1-h_\Theta(x)) \text{ if } y=0$$

Equation 1

1.4.1.7 Gradient descent

Gradient Descent is applied like before in linear Regression in order to minimize the cost. Conjugate gradient like BFGS is also other more well versed optimization algorithms out there.

1.4.2 Random forest

Random forest is made up of many decision trees and is an intuitive model. The decision tree is a hierarchical series of yes/no questions asked about our data gradually leading to a predicted class (or continuous value in the case of regression). It is an interpretable model as it makes classifications much we usually do. There are sequences of queries regarding the given data that we have until we conclude at a decision (in an ideal world).

Random forest is a supervised learning algorithm which is used for both regression as well as classification. But it is mainly used for classification problems. As we already seen that a forest is made up of trees and more trees indicates more robust forest. Similarly, random forest algorithm produce decision trees on data samples and then receives the prediction from each of them and at last selects the best solution by the way of voting. It is an ensemble method which is better than a single decision tree because the over-fitting problem is reduced by averaging the result. Figure 4 shows that entire training sample is split into n training samples each of these samples represent a tree, which cumulatively form a random forest.

1.4.2.1 Random Forest pseudo code

Step 1: Randomly take “k” Characteristics from total “m” characteristics.

Where k << m

Step 2: Using calculation identify the node “d” using the best split point, from the “k” features,

Step 3: The Daughter nodes are split from existing nodes using the best split.

Step 4: Repeat 1 to 3 steps until number of nodes reached is ‘1’.

Step 5: By repeating steps 1 to 4 for “n” number times the forest is built that has “n” number of trees.

Initially the random forest algorithm starts by randomly selecting “k” features out of total “m” features. From the figure 4, it is obvious that we are randomly taking features and observations.

In the subsequent stage, we are taking the randomly selected “k” features to identify the root node by using the best split approach.

For the upcoming stage, the daughter nodes are calculated using the same best split approach. With the first 3 stages, we form the tree with a root node and having the target as the leaf node.

Finally, we repeat 1 to 4 stages to create “n” randomly created trees. The random forest is formed using these randomly created trees.

1.4.2.2 Random forest prediction code

For performing prediction using the trained random forest algorithm the following pseudo code is used.

1. The test features are taken and the rules are used for each randomly created decision tree to predict the outcome and the predicted outcome is stored.
2. The votes for each predicted target is calculated.
3. Assume the predicted target that got high vote as the final prediction from the random forest algorithm.

In order to perform the prediction using the trained random forest algorithm, the test features are passed through the rules of each randomly created tree. Consider we formed 100 random decision trees to from the random forest.

Each random forest will predict different target for the same test feature. By considering each predicted target, votes will be calculated. As said the 100 random decision trees are prediction of some 3 unique targets a, b, c then the votes of a is nothing but out of 100 random decision tree that is how many trees prediction is ‘a’.

Similarly for other 2 outcomes (targets) that is b, c. Now If ‘a’ is getting high number of votes that is assume that out of 100 random decision trees almost 65 trees are predicting the target will be as ‘a’.

Then the final random forest returns ‘a’ as the predicted target. This method of voting is known as majority voting.

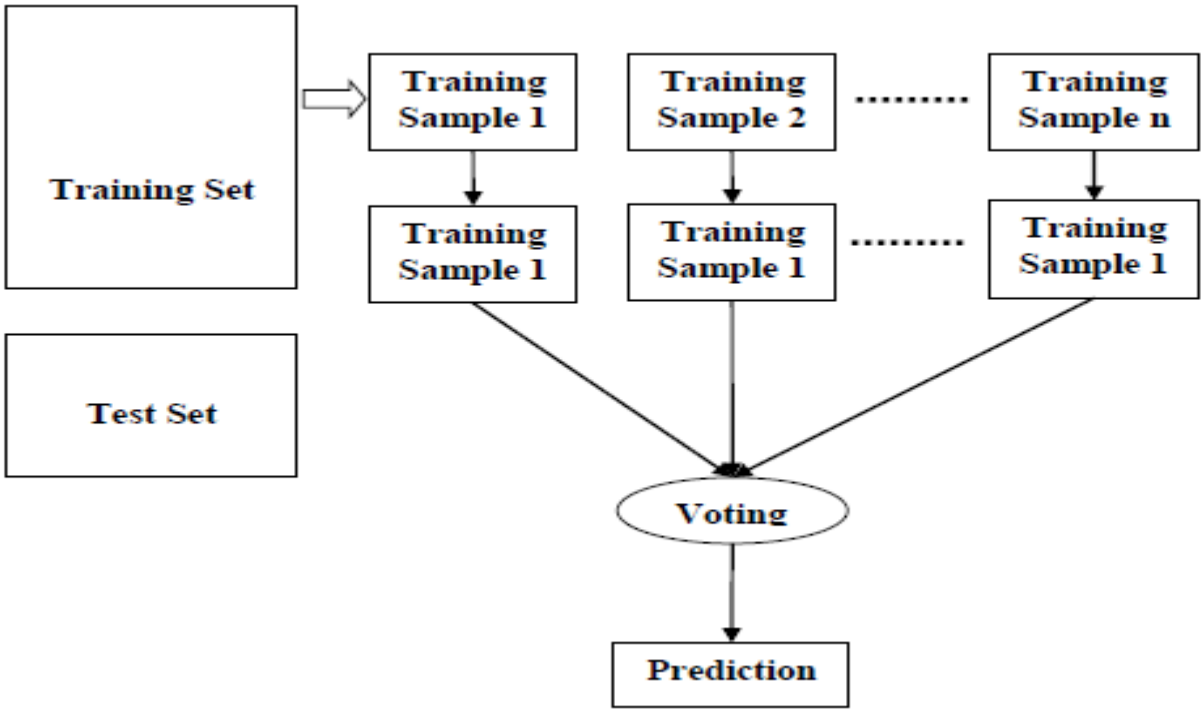


Figure 4 Random Forest

1.4.3 ADABOOST Algorithm:

1.4.3.1 Boosting Ensemble Method

Boosting is a general ensemble method that produces a strong classifier from many numbers of weak classifiers.

This is obtained by creating a model from the training data, then creating another (second) model that tries to correct the errors that occurred in the first model.

Models are further added until the training set is predicted perfectly or a maximum number of models for addition reach.

AdaBoost was the first really successful boosting algorithm developed for binary classification.

1.4.3.2 Adaboost Algorithm

AdaBoost is best algorithm that is used to boost the effectiveness of decision trees on binary classification problems.

AdaBoost was originally called as AdaBoostM1 by the authors of the technique Schapire and Freund. More recently it is referred as discrete AdaBoost because it is used for classification rather than regression.

The most suited and therefore most usual algorithm used with AdaBoost is a decision trees with one level. Because these trees are so short and only contain one decision for classification, they are mostly called as decision stumps.

Any machine learning algorithm performance can be improved using AdaBoost algorithm. It is best worked with weak learners. These are models that achieve high accuracy just above random chance on a classification problem. The final equation for classification can be represented as

$$F(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x)),$$

Equation 2

Where, f_m stands for the m_th weak classifier

theta_m is the corresponding weight

It is accurately the weighted combination of M weak classifiers. The entire process of the AdaBoost algorithm can be explained as follow.

1.4.3.3 AdaBoost algorithm

Given a dataset containing n points, where

$$x_i \in \mathbb{R}^d, y_i \in \{-1, 1\}.$$

Here -1 represents the negative class and 1 represents the positive class.

Initialize the weight for each data point as:

$$w(x_i, y_i) = \frac{1}{n}, i = 1, ..., n.$$

1.4.3.4 For iteration m=1,…,M

(1) Fit weak classifiers to the data set and select the one with the lowest weighted classification error:

$$\epsilon_m = E_{w_m}[1_{y \neq f(x)}]$$

(2) Calculate the weight for the m_th weak classifier:

$$\theta_m = \frac{1}{2} ln(\frac{1 - \epsilon_m}{\epsilon_m}).$$

Any classifier that has accuracy more than 50%, the weight is positive. The larger the weight, when the classifier is more accurate. The weight is negative for

the classifier with less than 50% accuracy. We combine its prediction by flipping the sign. For example, a classifier with 40% accuracy can be turned into 60% accuracy by flipping the sign of the prediction. Thus even the classifier performs worse than random guessing, it still contributes to the final prediction. We only don't want any classifier with exact 50% accuracy, which doesn't add any information and thus contributes nothing to the final prediction.

(3) Update the weight for each data point as:

$$w_{m+1}(x_i, y_i) = \frac{w_m(x_i, y_i) exp[-\theta_m y_i f_m(x_i)]}{Z_m},$$

where Z_m is a normalization factor that takes care that the sum of all instance weights is equal to 1.

If a misclassified case is from a positive weighted classifier, the "exp" term in the numerator would be always larger than 1 that is y*f is always -1, theta_m is positive. After iteration, the misclassified cases will be updated with larger weights. The same logic applies to the negative weighted classifiers. The only difference is that the original correct classifications would become misclassifications after flipping the sign.

After M iterations is over we will get the final prediction by summing up the weighted prediction of each classifier.

CHAPTER 2

LITERATURE SURVEY

Malware is harmful software, expanding on computer linked architecture, hat is designed to damage computer system without the owner's knowledge of the system and technological advances posing major challenges for researchers in academic and industry. This malicious activity that can distribute malware in different forms through advertising. The purpose of the study is to examine the available literature on malware analysis and to determine how research has evolved and advanced in terms of the size, content and finally publishing .Malware advertising, revenue generation and various internet companies elaborate advertising systems. For example goggle, yahoo, Microsoft. The malware detection system is used to identify malicious actions. If a malware is introduced with a new signature it becomes difficult to detect if it's malicious. Signature-based detection is not so effective in these days. The malware attacks can exploit the setting until a signature is created for a new malware. Antimalware software are capable of detecting previously unknown malware and zero day attacks. Malware detection is an important factor in the security of the computer systems. However the currently used signature methods do not accurately detect zero day attacks and polymorphic viruses. So, the need for machine learning based diagnostics arises. The purpose of this work is to determine the best precision extraction feature representation and classification methods .The dataset is used for this study contains 1156 malware files of 9 different types[1].

Malware, a software disease, is becoming chronic. Technology must be used to analyze the advantage and disadvantage analytically. The purpose of this paper

is to analyze the existing publication in the regard, following the trend in progress [2].

Research shows that, malware has grown exponentially, causing financial loss to several industries. A lot of antimalware industries have come up with solutions to prevent attacks from this malware. The speed size and complexity of malware pose challenges to the antimalware community. Recently, researches have begun to use machine learning and deep learning methods for malware detection [3].

Malware is any program that could cause harm to the computer system. The existing network of malware includes Ransom ware, adware, key loggers, viruses, Trojan horse, worms and others. The growth of malware promotes a great risk to the security of confidentiality information. There is an immediate need to determine the effectiveness of available machine learning classification algorithms used for malware detection. Malicious programming, shortly known as malware, can be physically conveyed to a USB device or other configuration. Malware is under serious security threads, which are sophisticated with advanced techniques, such as polymorphism, that make it difficult to detect and analyze the results [4].

Static and dynamic techniques have been developed to detect malware, classifying them into malware families. Malware identification has advantages over malware detection, because it’s difficult to hide the behavior of malware, when operating under standard malware detection [5].

Detecting and classifying malware is an important task in the data processing industry. Malware can easily damage the computer by causing harm to the user’s computer so some techniques have been developed for accurate malware detection. Some lacks are misdirection and high performance, more time required

to classify malware, increasing the complexity. We present a framework that can use different machine learning algorithms to successfully distinguish malware files and also cleaning them. In this paper we present the concept behind our framework first, later testing it in a medium sized data set of malware and clean files. Traditional malware detection cannot effectively detect this new decade malware so article proposes a model to create malware datasets that detects semantically related behaviors from sample programs. Features that do not exceed the specified score will be removed from the dataset. The detection rate and the accuracy of the proposed model are higher than the known models [6].

Machine learning is an essential part of the cyber security, where huge amount of data is collected and processed to provide security solutions, that enables to detect and analyze malware variants without needing extensive resources [7].

Data mining techniques for malware detection have gathered together over decades, increasing its complexity and hence the proposed mechanisms are inadequate making it difficult to recognize. This paper presents systematic and comprehensive survey of malware detection methods using data mining techniques [8].

CHAPTER 3
METHODOLOGY

3.1 Proposed Methodology

Generally traditional methods like signature based methodology are used to detect the malwares. But machine learning approaches seem to be more effective than traditional methods. Since there are lots of algorithms available we choose only three algorithms and compared their performance. Among three algorithms, Random forest seems to outperform the other two algorithms.

3.2 WEKA

WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from our own Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

3.3 Dataset

The data is obtained from the UCI repository website. The dataset contains the dynamic feature of 107,888 executables collected by Virus Share from Nov/2010 to Jul/2014. Figure 5 shows the number of malicious executables collected each year in a graphical representation. The features were extracted from the artifacts generated by the executables in the Cuckoo sandbox. VirusTotal.com is an online service which analyzes files and URL's for malicious content such as virus, worm and Trojan by leveraging on an array of 52 commercial antivirus solutions for the detection of malicious signatures.

On record, VirusTotal receives and analyzes nearly 2 million files every day. There are 45 text files in the total dataset.

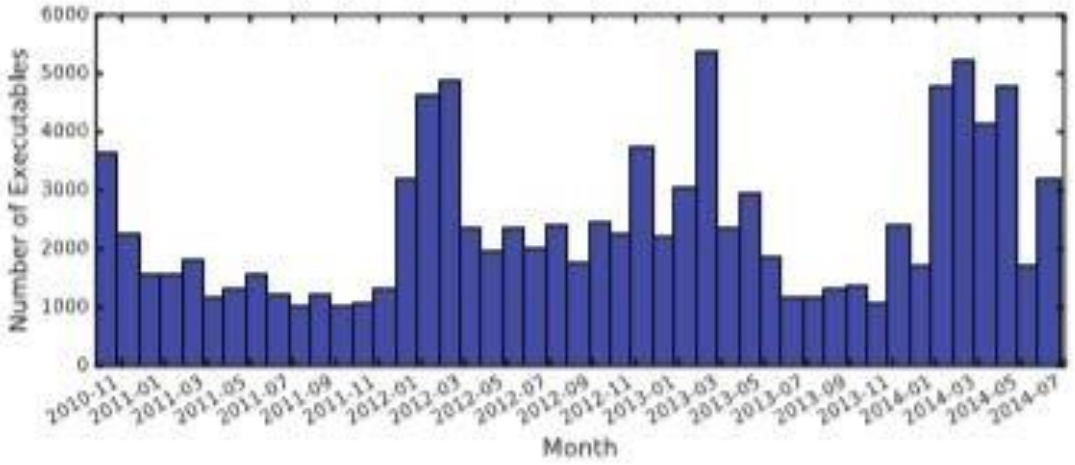


Figure 5 Distributions of Executables

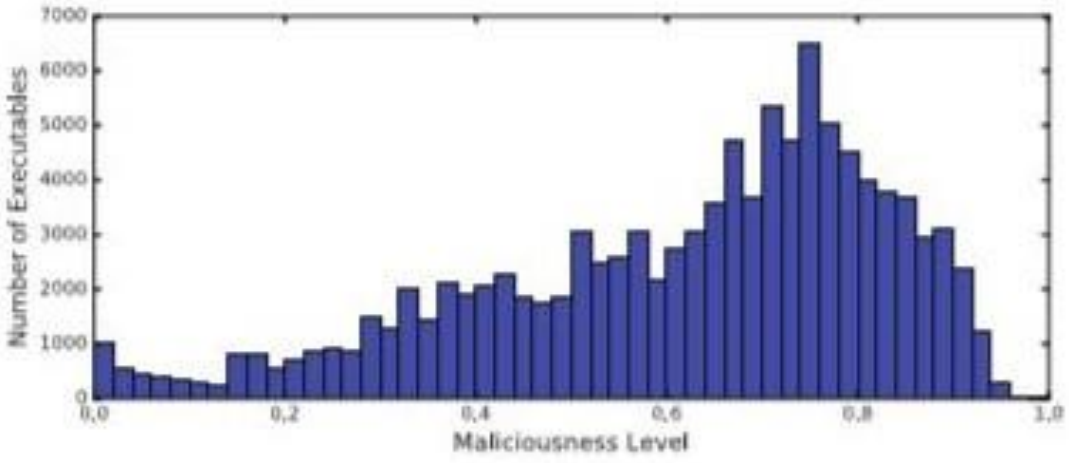


Figure 6 Distribution of risk level

Initially, the text files are not in a structured format. It has the risk values along with the features. There are 483 features mentioned, not all the files have all the features. The risk value and features of a file is mentioned in each row. Figure 6 mention the risk levels of the executables.

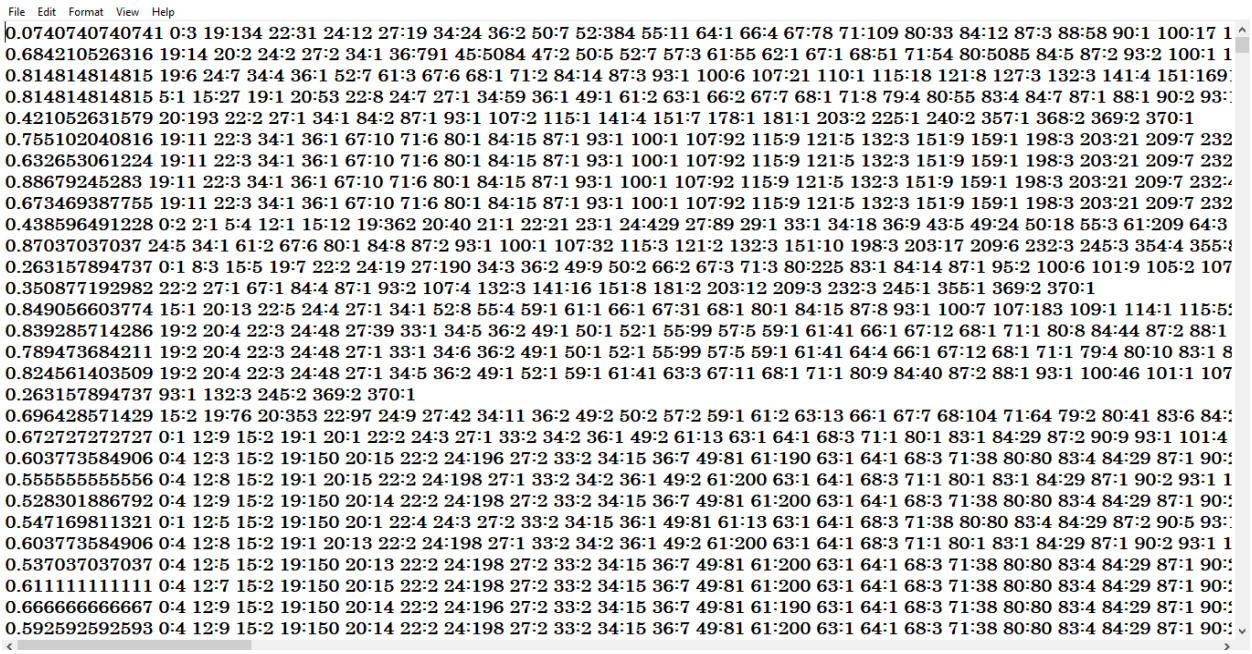
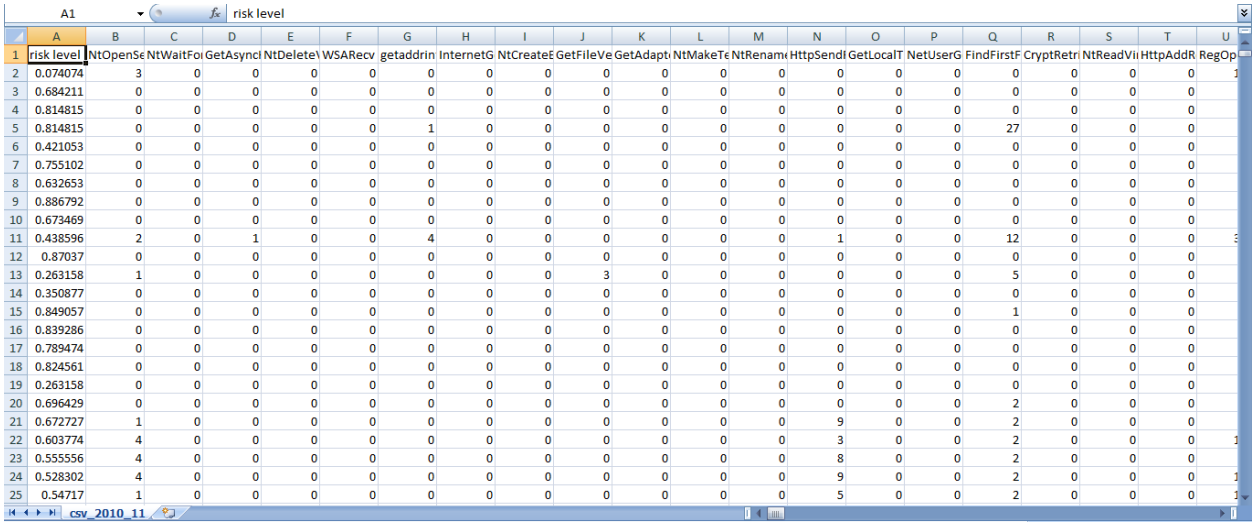


Figure 7 Unstructured format of data file

Python code is used to read the unstructured text file as shown in Figure 7 and it is converted into CSV format as shown in Figure 8. Each text file contains almost 3000 features and risk values of executable files.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	risk level	NtOpenSe	NtWaitFo	GetAsyncI	NtDeleteV	WSARecv	getaddrin	InternetG	NtCreateE	GetFileVe	GetAdapt	NtMakeTe	NtRename	HttpSendI	GetLocalT	NetUserG	FindFirstF	CryptRetr	NtReadVi	HttpAddR	RegOp
2	0.074074	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	0.684211	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0.814815	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0.814815	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	27	0	0	0	
6	0.421053	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0.755102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0.632653	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0.886792	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0.673469	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0.438596	2	0	1	0	0	4	0	0	0	0	0	0	1	0	0	12	0	0	0	
12	0.87037	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0.263158	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	5	0	0	0	
14	0.350877	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0.849057	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
16	0.839286	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0.789474	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0.824561	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19	0.263158	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20	0.696429	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	
21	0.672727	1	0	0	0	0	0	0	0	0	0	0	0	9	0	0	2	0	0	0	
22	0.603774	4	0	0	0	0	0	0	0	0	0	0	0	3	0	0	2	0	0	0	1
23	0.555556	4	0	0	0	0	0	0	0	0	0	0	0	8	0	0	2	0	0	0	
24	0.528302	4	0	0	0	0	0	0	0	0	0	0	0	9	0	0	2	0	0	0	1
25	0.54717	1	0	0	0	0	0	0	0	0	0	0	0	5	0	0	2	0	0	0	1

Figure 8 Structured format of data file

CHAPTER 4
RESULTS AND DISCUSSION

In this project, we have performed analysis on Malware detection.For this we have used a machine learning algorithms called Logistic Regression,Random forest and Adaboost.

These data sets contain 45 months of executable files that are collected from the virushare websites. The range of data is from November 2010 to July 2014.

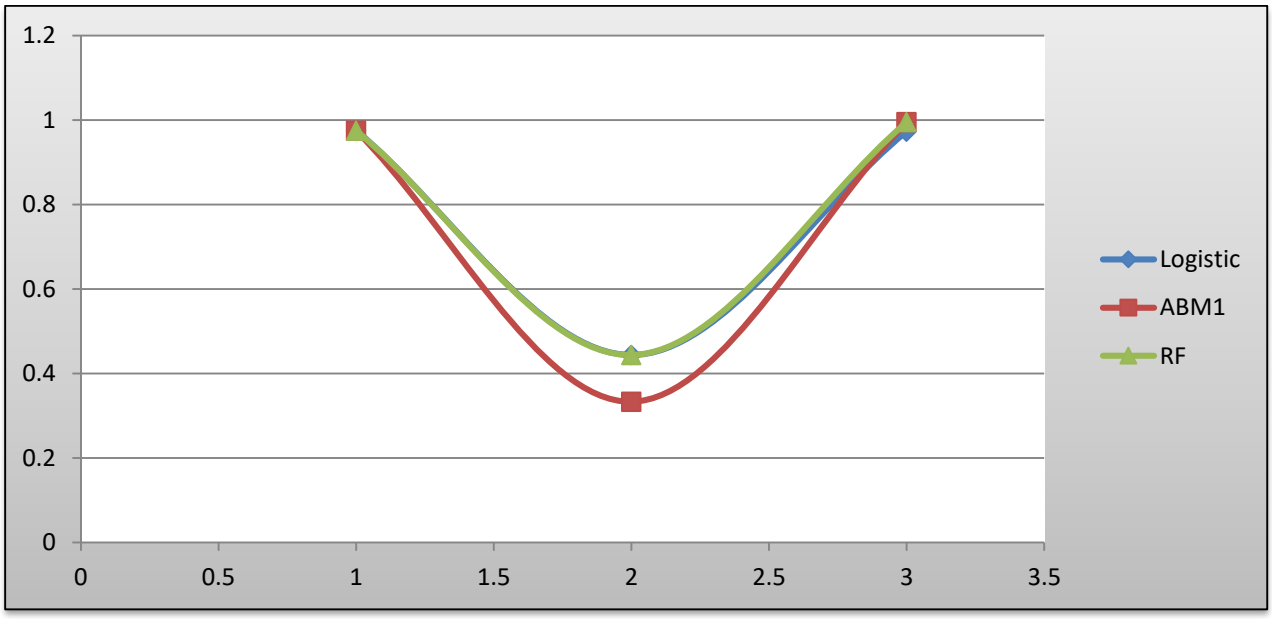


Figure 9 TP Rate (FOLD_ Value = 5)

True positive is the measure the percentage of actual positives which are correctly identified. Figure 9 shows the true positive rate of all the three machine learning algorithms. In which logistic regression and random forest almost show the same output.

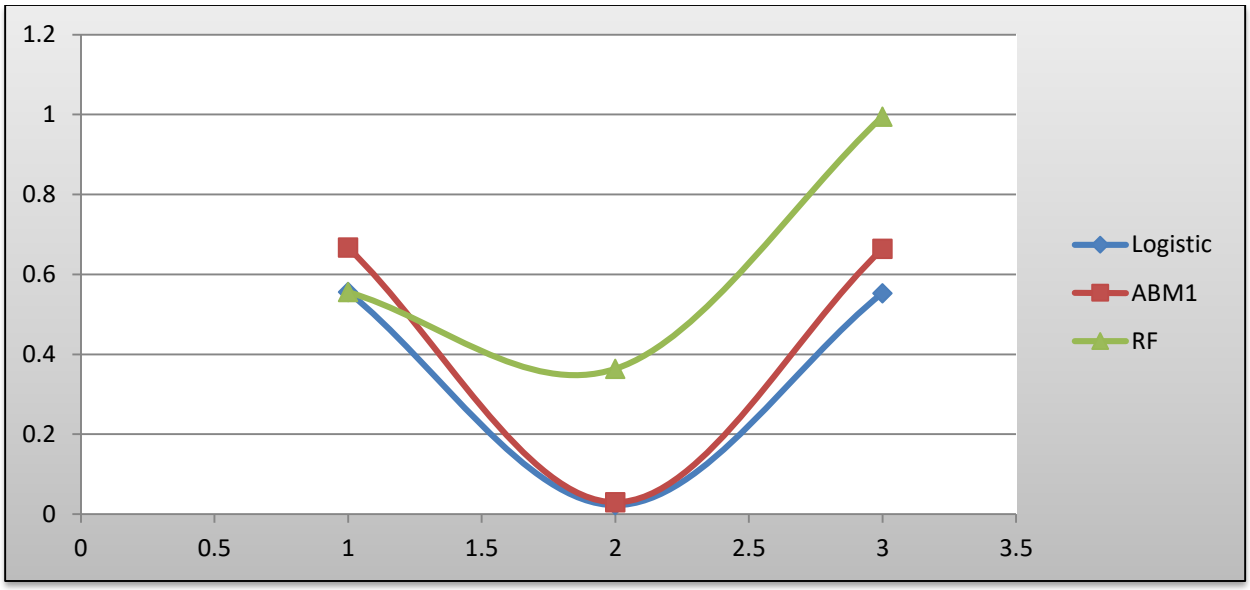


Figure 10 FP Rate (FOLD_ Value = 5)

False positive rate is the measure of incorrect prediction of positives, that is the positives are predicted as negatives. From Figure 10 it is clear that random forest shows high false positive rate than other two algorithms.

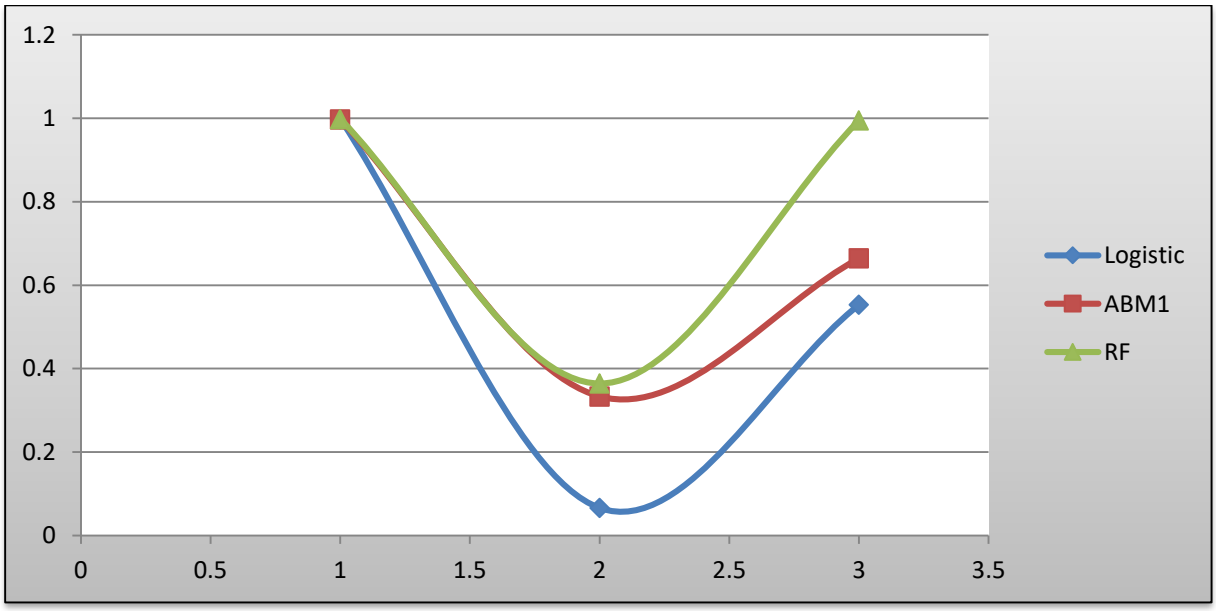


Figure 11 Precision (FOLD_ Value = 5)

Precision is used to evaluate the performance of classification or information retrieval systems. The fraction of relevant instances among all retrieved instances is called Precision. Figure 11 show that Random forests have higher precision for the given set of data.

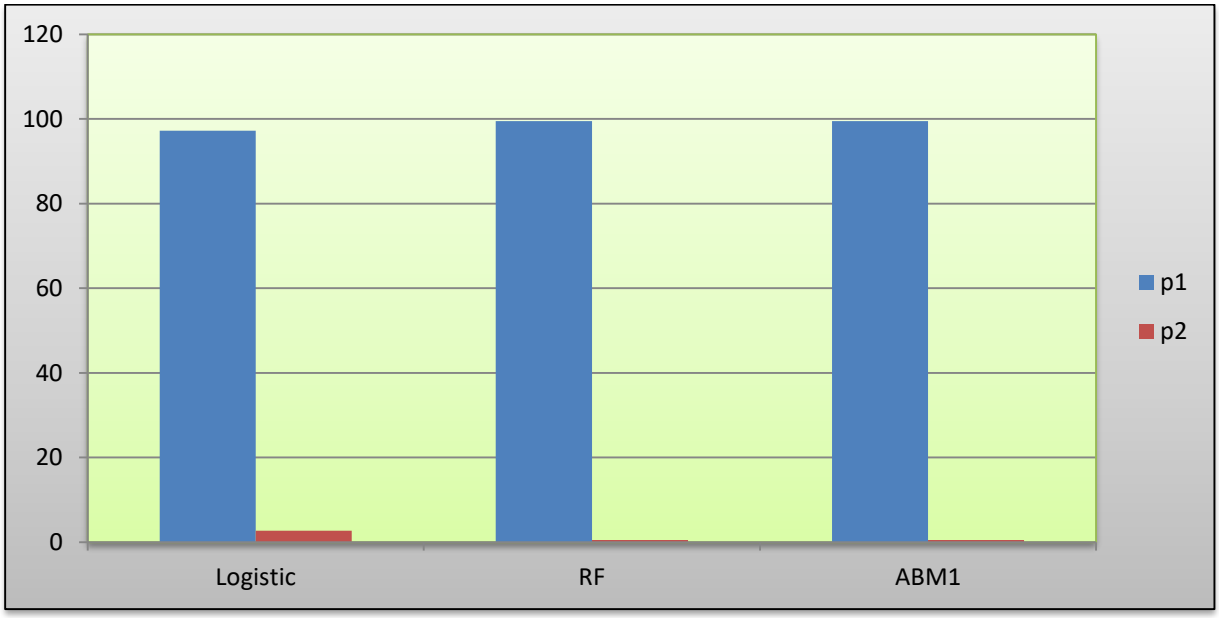


Figure 12 Correctly Classified and incorrectly classified (FOLD_ Value = 5)

In figure 12, the p1 and p2 represent the correctly classified and incorrectly classified. The Random forest and AdaBoost algorithm have almost all the samples that are correctly classified.

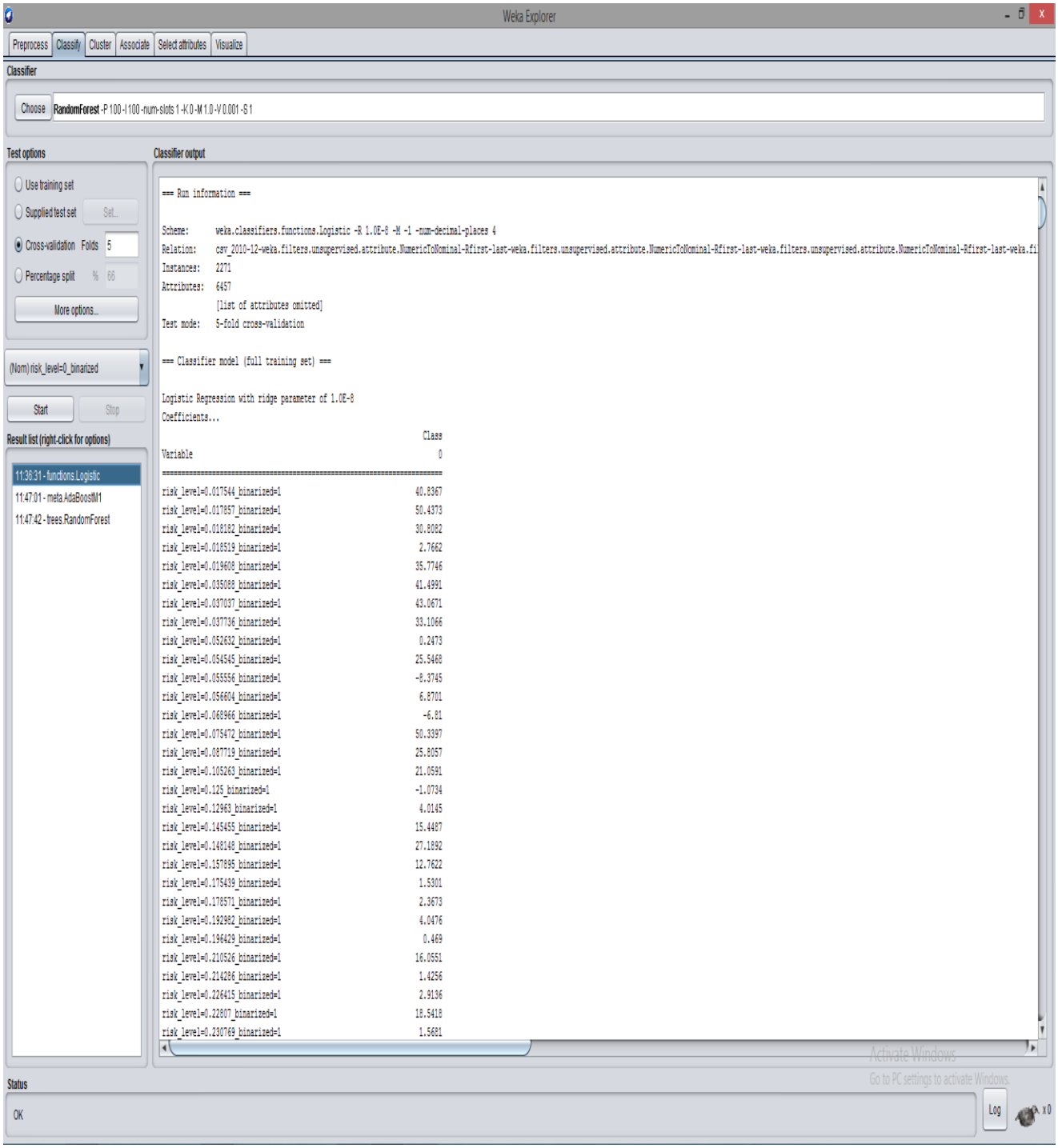


Figure 13 Training result in Logistic Regression

Figure 13 shows the output of training of logistic regression with rigid parameter of 1.0E -8 .The training set has 2271 instances and 6457 attributes.

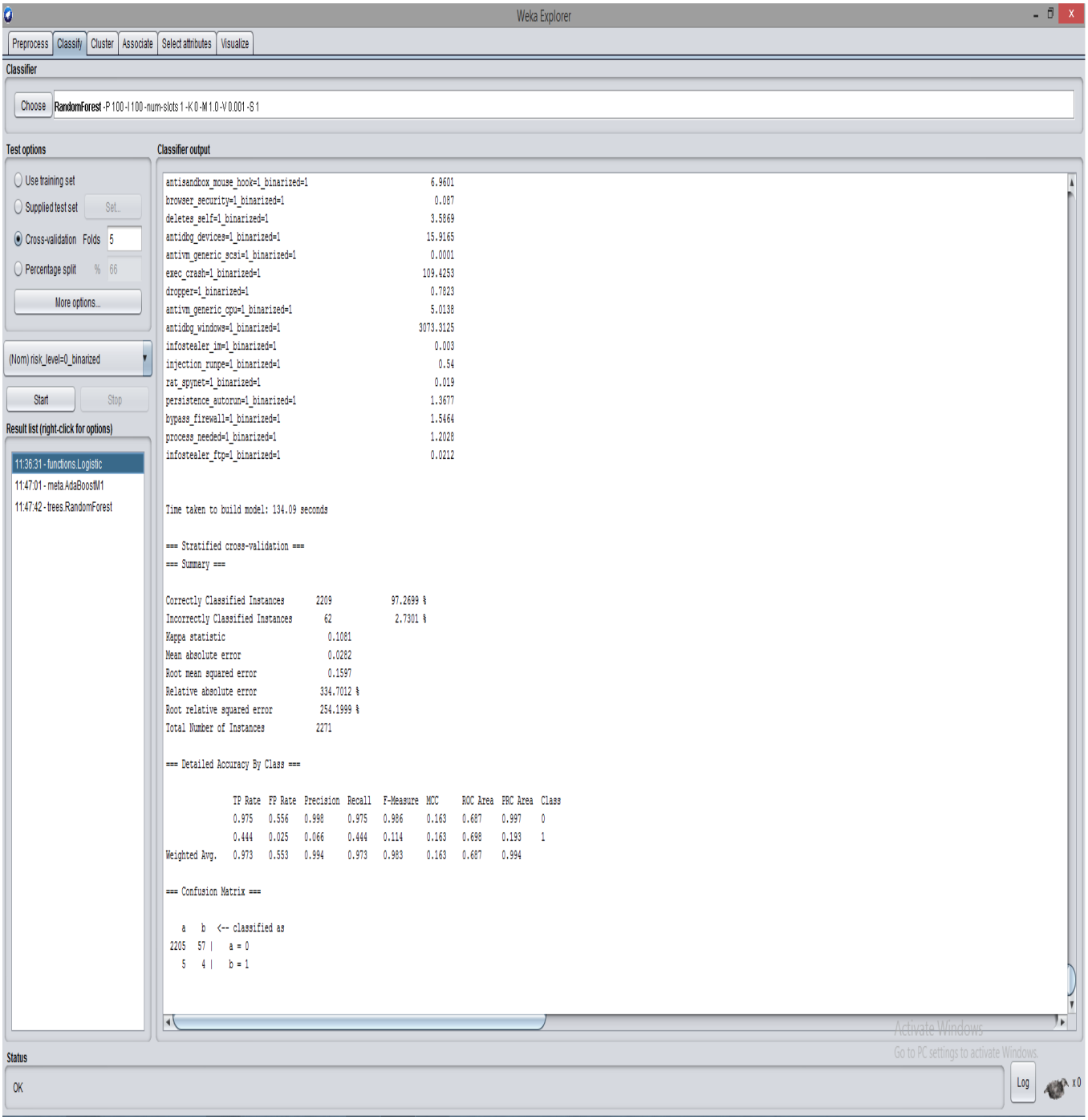


Figure 14 Testing result in Logistic Regression

The testing output of logistic regression is shown in figure 14. The values 97.2699 % and 2.7301 % are correctly and incorrectly classified values respectively. It has a precision value of about 0.998

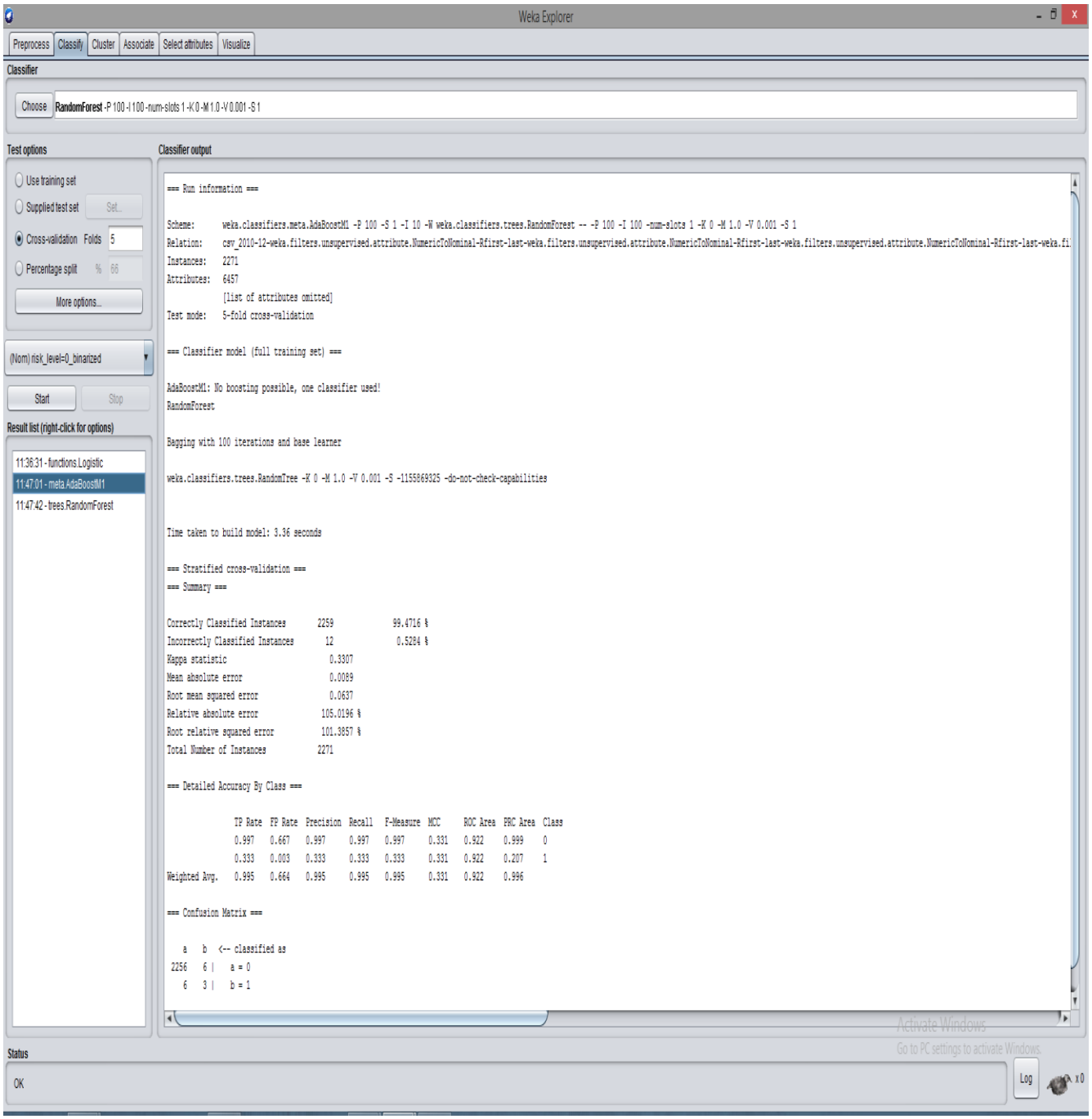


Figure 15 Training and Testing result in AdaBoost

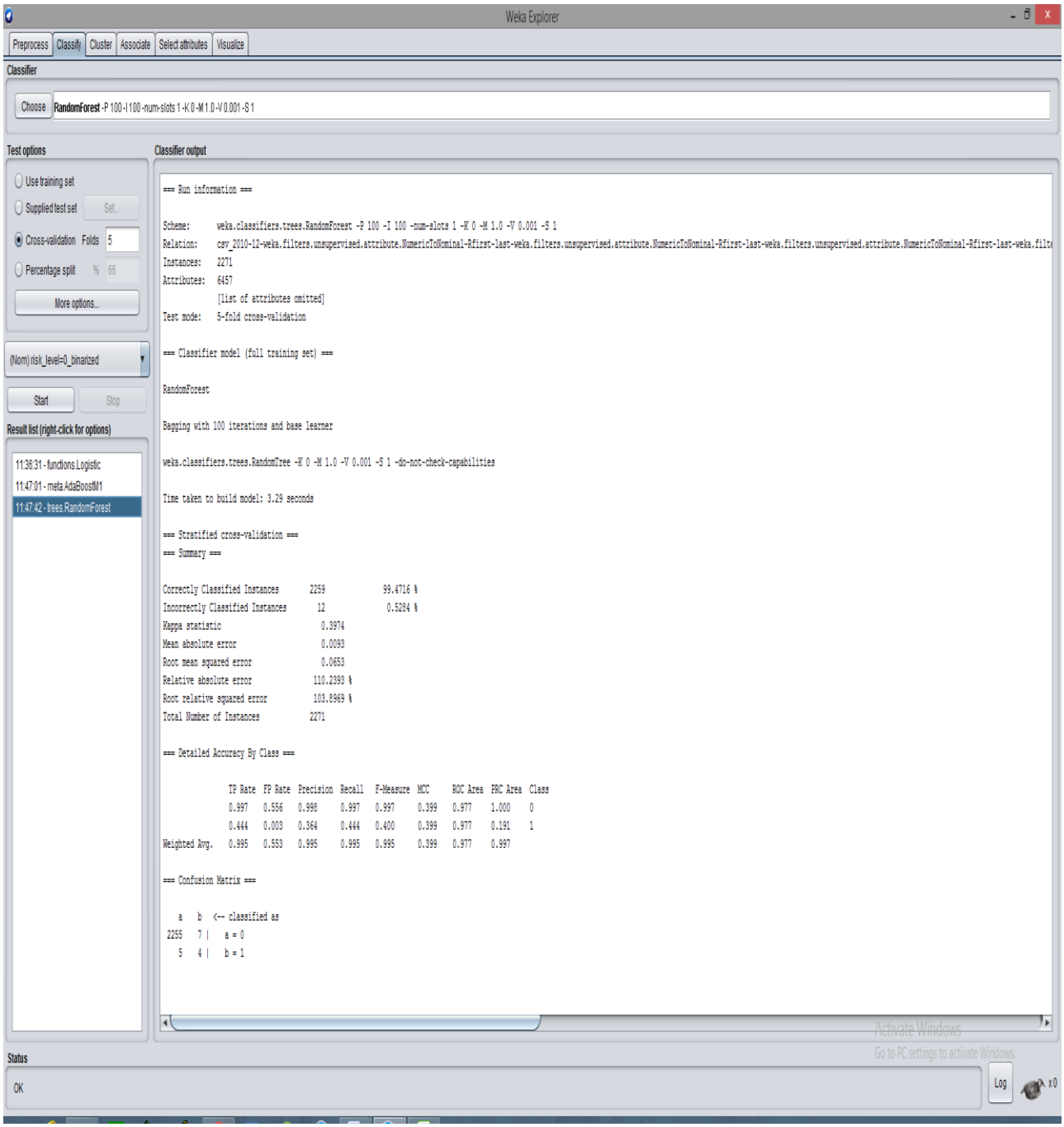


Figure 16 Training and testing result in Random Forest

As shown in Figure 15 and 16 shows the training and testing of Adaboost algorithm and Random forest respectively.

CHAPTER 5
CONCLUSION

As malwares are evolving rapidly, it is difficult to detect them. In this project, The dataset that has nearly 10 lakh instances which is collected from virushare.com. Here Adaboost, Random forest, logistic regression algorithms are used to detect the malwares. Among three of them, Random forest is more efficient considering the precision value. It is followed by Adaboost algorithm and it is further followed by Logistic regression. There are various characteristics of the executable malwares file included. They are grouped into four categories namely API Call Category, Registry Category, File System Category, Miscellaneous Category. Over 483 attributes are taken into account.

The project can be investigated in many ways. The first way is to explore effective methods other than what we have used which might give better performance on malware detection and the second way is to use different kinds of parameters other than we have used.

APPENDIX

Source Code

```
import json
from collections import defaultdict
from pprint import pprint
import csv
f1 = open("2010-11.txt", "r")
Final_list = []
sub_list = ["#"]
for i in range(0, 483):
    sub_list.append(i)
Final_list.append(sub_list)
row = 1
f1.seek(0)
count=0
for line in f1:
    sub_list=[]
    D = dict()
    S = line.split ()
    hashcode=float(S[0])
    sub_list.append(hashcode)
```

```
    Zero=[0]*483
    sub_list.extend(Zero)

    for i in range(1,len(S)):
        ind=S[i].index(":")
        key=int(S[i][:ind])
        value=int(S[i][ind+1:])
        index=key+1
        sub_list[index]=value

    Final_list.append(sub_list)

f2=open("new_2010_11.txt","w")
f3=open("csv_2010_11.csv","w",newline='')
writer=csv.writer(f3)
for row in Final_list:
    count=count+1
    if count==1:
        continue
    line=str()
    for i in range(0,len(row)):
        str1=str(row[i])+" "
```

```
        line=line+str1
    f2.write(line.rstrip()+"\n")
    writer.writerow(row)
f2.close()
f3.close()
```

REFERENCES

[1] T. An and M. Kim, "A New Diverse AdaBoost Classifier," 2010 International Conference on Artificial Intelligence and Computational Intelligence, Sanya, 2010, pp. 359-363, doi: 10.1109/AICI.2010.82.

[2] E. Aydoğan and S. Şen, "Analysis of machine learning methods on malware detection," 2014 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, 2014, pp. 2066-2069, doi: 10.1109/SIU.2014.6830667.

[3] Bekerman, D., Shapira, B., Rokach, L., Bar, A.: Unknown malware detection using network traffic classification. In: IEEE Conference on Communications and Network Security (CNS), pp. 134–142 (2015)

[4] S. Choi, S. Jang, Y. Kim and J. Kim, "Malware detection using malware image and deep learning," 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2017, pp. 1193-1195, doi: 10.1109/ICTC.2017.8190895.

[5] D. Gavriluţ, M. Cimpoeşu, D. Anton and L. Ciortuz, "Malware detection using machine learning," 2009 International Multiconference on Computer Science and Information Technology, Mragowo, 2009, pp. 735-741, doi: 10.1109/IMCSIT.2009.5352759.

[6] T. Haifley, "Linear logistic regression: an introduction," IEEE International Integrated Reliability Workshop Final Report, 2002., Lake Tahoe, CA, USA, 2002, pp. 184-187, doi: 10.1109/IRWS.2002.1194264.

[7] A. Jain and A. K. Singh, "Integrated Malware analysis using machine learning," 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), Noida, 2017, pp. 1-8, doi: 10.1109/TEL-NET.2017.8343554.

[8] J. K. Jaiswal and R. Samikannu, "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression," 2017 World Congress on Computing and Communication Technologies (WCCCT), Tiruchirappalli, 2017, pp. 65-68, doi: 10.1109/WCCCT.2016.25.

[9] M. Liu, G. Wu, S. Wen and J. Chen, "An improved face detection classifier based on AdaBoost algorithm," 2011 4th International Congress on Image and Signal Processing, Shanghai, 2011, pp. 85-89, doi: 10.1109/CISP.2011.6099968.

[10] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha and S. Kundu, "Improved Random Forest for Classification," in IEEE Transactions on Image Processing, vol. 27, no. 8, pp. 4012-4024, Aug. 2018, doi: 10.1109/TIP.2018.2834830.

[11] S. V. Patel and V. N. Jokhakar, "A random forest based machine learning approach for mild steel defect diagnosis," 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, 2016, pp. 1-8, doi: 10.1109/ICCIC.2016.7919549.

[12] X. Shu and P. Wang, "An Improved Adaboost Algorithm Based on Uncertain Functions," 2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration, Wuhan, 2015, pp. 136-139, doi: 10.1109/ICIICII.2015.117.

[14] Y. Zhang et al., "Research and Application of AdaBoost Algorithm Based on SVM," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2019, pp. 662-666, doi: 10.1109/ITAIC.2019.8785556.

[15] J. Zhao, S. Zhang, B. Liu and B. Cui, "Malware Detection Using Machine Learning Based on the Combination of Dynamic and Static Features," 2018 27th International Conference on Computer Communication and Networks (ICCCN), Hangzhou, 2018, pp. 1-6, doi: 10.1109/ICCCN.2018.8487459.