INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
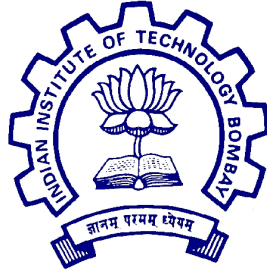
Project Report on

MALWARE DETECTION

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

CS725:Foundations of Machine Learning (Computer Science and Engineering)

BY

Anurag Chaudhary	Roll No:193050061
Himanshu Aswal	Roll No:193059001
Pranav Chaudhary	Roll No:193059004
Sanyam Raj	Roll No:193050096

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Project Entitled

MALWARE DETECTION
Submitted by

Anurag Chaudhary	**Roll No:193050061**
Himanshu Aswal	**Roll No:193059001**
Pranav Chaudhary	**Roll No:193059004**
Sanyam Raj	**Roll No:193050096**

is a bonafide work carried out by students and it is submitted towards the partial fulfillment of the requirement of CS725:Foundations of Machine Learning (Computer Science and Engineering).

Prof. Sunita Sarawagi
Department of Computer Science, IIT Bombay

Abstract

In recent years, the malware industry has become a well organized market involving large amounts of money. Well funded, multi-player syndicates invest heavily in technologies and capabilities built to evade traditional protection, requiring anti-malware vendors to develop counter mechanisms for finding and deactivating them. In the meantime, they inflict real financial and emotional pain to users of computer systems.

One of the major challenges that anti-malware faces today is the vast amounts of data and files which need to be evaluated for potential malicious intent. For example, Microsoft's real-time detection anti-malware products are present on over 160M computers worldwide and inspect over 700M computers monthly. This generates tens of millions of daily data points to be analyzed as potential malware. One of the main reasons for these high volumes of different files is the fact that, in order to evade detection, malware authors introduce polymorphism to the malicious components. This means that malicious files belonging to the same malware "family", with the same forms of malicious behavior, are constantly modified and/or obfuscated using various tactics, such that they look like many different files.

In order to be effective in analyzing and classifying such large amounts of files, we need to be able to group them into groups and identify their respective families. In addition, such grouping criteria may be applied to new files encountered on computers in order to detect them as malicious and of a certain family.

Contents

1 Problem Statement

In the past few years, the malware industry has grown very rapidly that, the syndicates invest heavily in technologies to evade traditional protection, forcing the anti-malware groups/communities to build more robust softwares to detect and terminate these attacks. The major part of protecting a computer system from a malware attack is to identify whether a given piece of file/software is a malware.

2 Goal and Objective

- Minimize multi-class error.

- Multi-class probability estimates.

- Malware detection should not take hours and block the user's computer. It should finish in a few seconds or a minute.

3 Data

Source : https://www.kaggle.com/c/malware-classification/data
        For every malware, we have two files
.asm file
.bytes file (the raw data contains the hexadecimal representation of the file's binary content, without the PE header).

Total train dataset consist of 200GB data out of which 50Gb of data is .bytes files and 150GB of data is .asm files.
There are total 10,868 .bytes files and 10,868 asm files total 21,736 files
There are 9 types of malwares (9 classes) in our given data:

- Ramnit
- Lollipop
- Kelihos_ver3
- Vundo
- Simda
- Tracur
- Kelihos_ver1
- Obfuscator.ACY
- Gatak
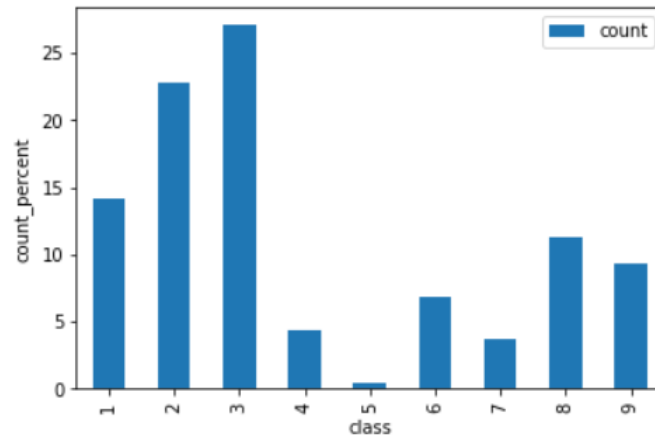
Figure 1: Data Distribution of various classes

4   Related Literature

- Microsoft Malware Winners' Interview: 1st place,"NO to overfitting!"
  http://blog.kaggle.com/2015/05/26/microsoft-malware-winners-interview-
  1st-place-no-to-overfitting/

- Novel Feature Extraction, Selection and Fusion for Effective Malware
  Family Classification
  https://arxiv.org/pdf/1511.04317.pdf

- First place approach in Microsoft Malware Classification Challenge
  (BIG 2015)
  https://www.youtube.com/watch?v=VLQTRlLGz5Y

- Malware Detection github
  https://github.com/dchad/malware-detection

5   Description of the set of approaches tried

- Logistic Regression- We first tried with the Logistic Regression Model and using this model, accuracy of 76% was achieved.

- Random Forest- Using this model, 96% accuracy was achieved.

6   Experiments

6.1   Code

The code is developed in Python with the help of libraries mainly matplotlib,numpy,pandas and sklearn.
The code started by first of all visualising the data distribution among various classes.
The code can be found on this link: `https://git.cse.iitb.ac.in/pranavchaudhary/CS725`

6.2   Experimental Platform

The code was developed and tested on Windows using Jupyter Notebook.
The code was run on a machine with configurations as:

- Processor: i7-9th Gen
- RAM: 16GB
- SSD: 256GB

The code ran on this machine for 60 hours(approx.)
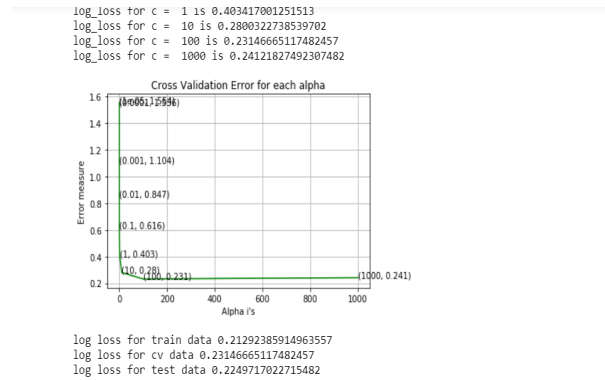
6.3 Experimental Results

```
log_loss for c =  1 is 0.403417001251513
log_loss for c =  10 is 0.2800322738539702
log_loss for c =  100 is 0.23146665117482457
log_loss for c =  1000 is 0.24121827492307482
```



```
log loss for train data 0.21292385914963557
log loss for cv data 0.23146665117482457
log loss for test data 0.2249717022715482
```

Figure 2: Logistic Regression Classifier Alpha vs. Loss Graph

```
log_loss for c =  10 is 0.047716095315105386
log_loss for c =  50 is 0.0428596306996971816
log_loss for c =  100 is 0.04369260598280953
log_loss for c =  500 is 0.04295321911674641
log_loss for c =  1000 is 0.04255114042507899
log_loss for c =  2000 is 0.04273066203790236
log_loss for c =  3000 is 0.04296630258270441
```



```
For values of best alpha =  1000 The train log loss is: 0.01644034189871256
For values of best alpha =  1000 The cross validation log loss is: 0.04255114042507899
For values of best alpha =  1000 The test log loss is: 0.03794906769143639
```
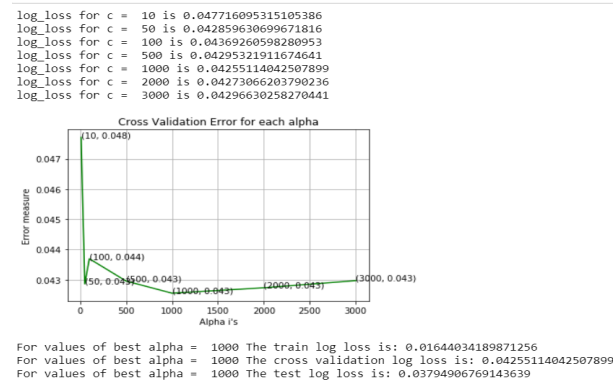
Figure 3: Random Forest Classifier Alpha vs. Loss Graph

7   Effort

The different parts of the project along with fraction of time taken by each part:

- Learning about Malwares and bytes and asm files-0.05

- Data Visualization-0.05

- Data Preprocessing-0.4

- Training-0.2

- Validation-0.2

- Testing-0.1

The most challenging and time taking part in this project was the pre-processing of dataset to a reasonable size without loss of information as the original dataset was large enough to train the model on our machines(around 184 GB). Fraction of work done by different team members:

- Anurag Chaudhary-0.25

- Himanshu Aswal-0.25

- Pranav Chaudhary-0.25

- Sanyam Raj-0.25

8 Conclusion

The dataset provided by Microsoft was of a very large size and had to be preprocessed using Feature Extraction to bring it to a size which could be run on our machines. The model was trained on bytes files as well as asm files using Logistic Regression Model and Random Forest Classifier Model. The results achieved by the Random Forest Model was much better as compared to the results achieved by the Logistic Regression Model.