Malware Detection using Machine Learning

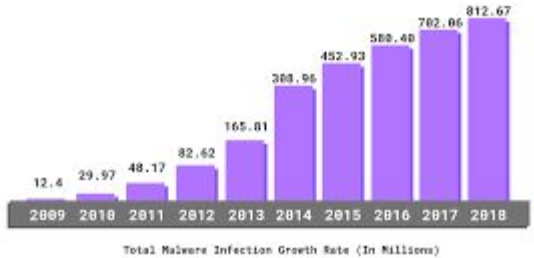
Harshita S - 1MS17IS047
Kritika Choudhary - 1MS17IS053
Lohith R - 1MS17IS057
Nikhil Kumar - 1MS17IS075

INTRODUCTION

Motivation

- The increase of malware that is exploiting the net daily has become a significant threat.
- Hackers build malware presentable to persuade users into downloading them.
- Users are unaware that the program is malware as it looks legitimate.
- Once installed, the malware hides in numerous folders within the system.
- Then it starts to encrypt files and record personal data.
- The malicious nature of the software can easily get access to the services provided by the device and collect sensory data and personal information.
- That's how malware gets downloaded on the system.

- According to Traficom, since 2018, worldwide malware attacks have risen by 350% in total.
- The Figure below shows the growth of malware from 2009-18.



- By using malware analysis combined with data processing tasks, such as, machine learning (classification) techniques to attain effectiveness and potency in detection malware.
- Acts as an early warning system.
- Hence, malware detection is considered as a profound solution.

Scope

- To perform classification, feature extraction and apply machine learning techniques to detect malware.
- To determine the best feature representation and feature extraction
- Malware removal or prevention will not be discussed.
- This study only focuses on malware detection.
- A pre-built dataset with the extracted PE features has been used.
- Results of various automated classification methods based on machine learning and compares them.

Objectives

- Antivirus software cannot fulfill the need for protection.
- The root cause consists within the software and can be fetched using malware detection techniques.
- To gain a deeper understanding of malwares and the techniques that are currently used for their detection.
- To understand machine learning algorithms using a practical approach.
- The machine learning process appraises the flexibility of the features that are to be chosen for the working of the project.
- The best feature elaborates on the closest accurate rate of malware detection.
- Features may vary for the benefit of the detection
- The right feature will complete the need for the algorithm to attain the best results

Proposed Model

- Current anti viruses still depend on signature-based methods for malware detection.
- Current detection methods are slowly becoming obsolete.
- The purpose of this project is to build a model that can predict whether a given Windows Portable Executable file is malware or benign statically using five machine learning algorithms.
- We then perform a comparative analysis of the algorithms using accuracy as the measure and determine the best one.

Reasons for choosing Portable Executable format :

- Windows is the most common operating system
- Collecting benign executables for Windows is easy
- The usability of methods for detection based on PE header attributes.
- Third reason is the detailed documentation of this file format.
- Around 47.80% of files submitted to Virustotal are PE files

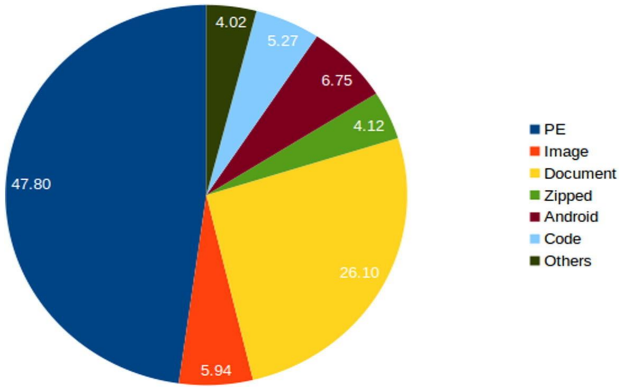
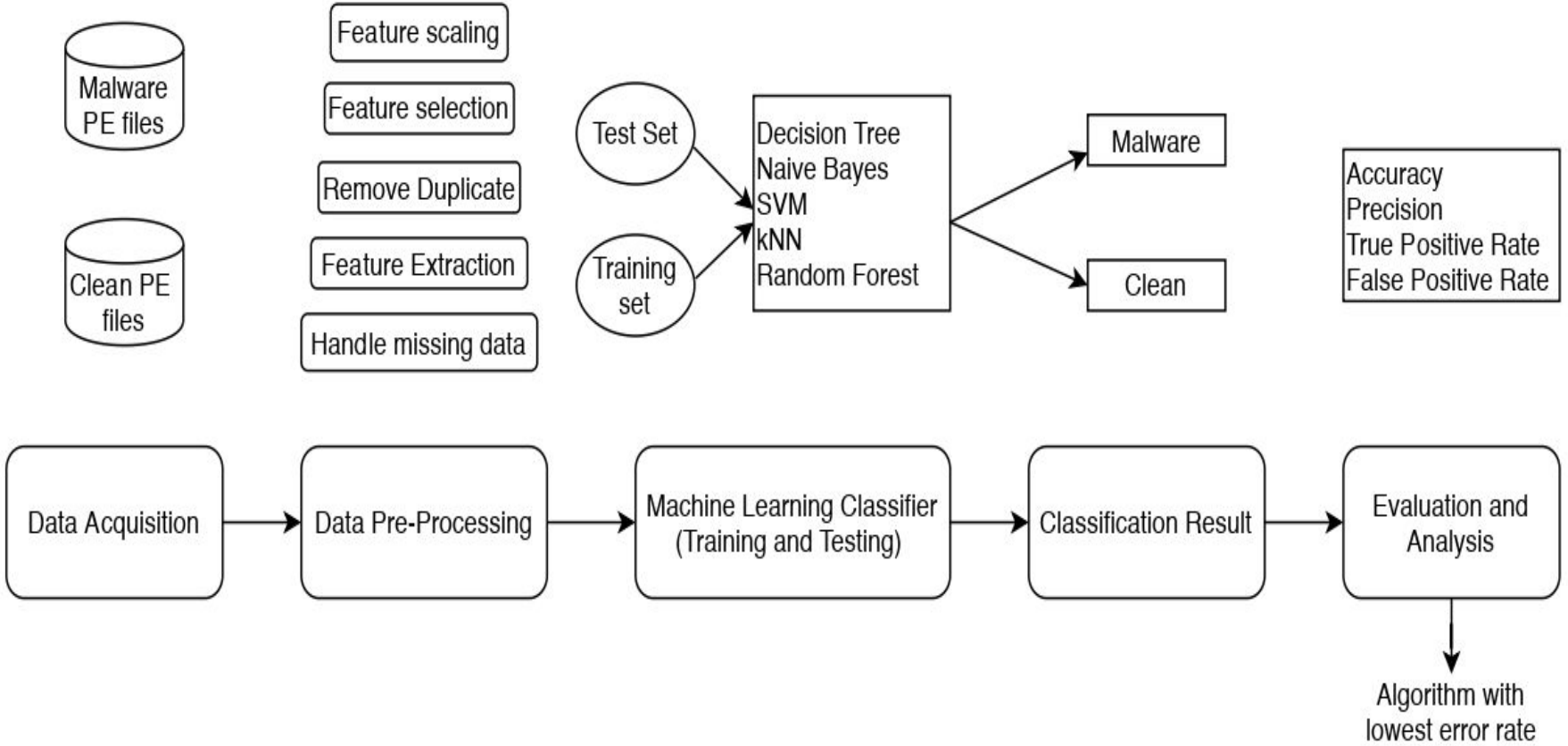


Fig. 1. Percentage of file type submitted to Virustotal (29 Oct- 4 Nov, 2016).

SYSTEM ARCHITECTURE and DESIGN


Malware PE files
Clean PE files
Feature scaling
Feature selection
Remove Duplicate
Feature Extraction
Handle missing data
Test Set
Training set
Decision Tree
Naive Bayes
SVM
kNN
Random Forest
Malware
Clean
Accuracy
Precision
True Positive Rate
False Positive Rate
Data Acquisition
Data Pre-Processing
Machine Learning Classifier (Training and Testing)
Classification Result
Evaluation and Analysis
Algorithm with lowest error rate

Data Acquisition

PE FILES:

The PE file format is a data structure that contains the information necessary for the Windows OS loader to manage the wrapped executable code.

Data Preprocessing

Handle missing data

Feature Extraction

Remove duplicates

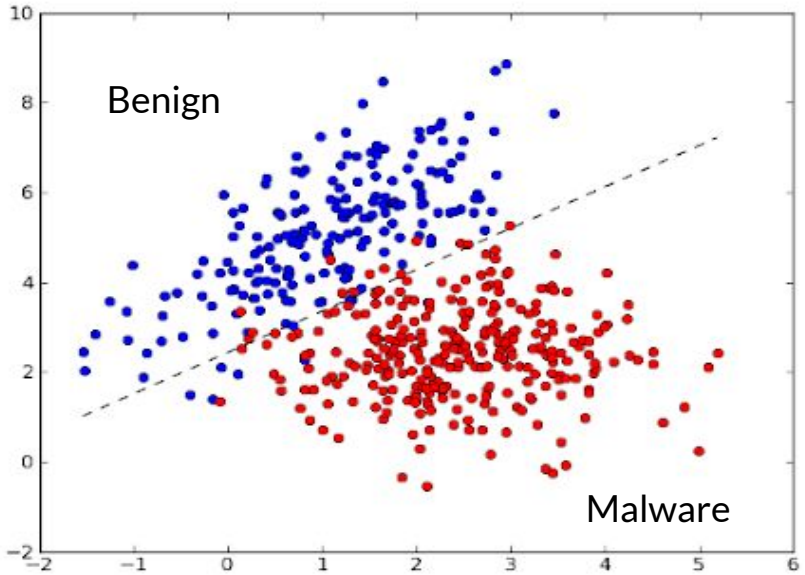
Standardize dataset

Splitting the dataset into train and test set

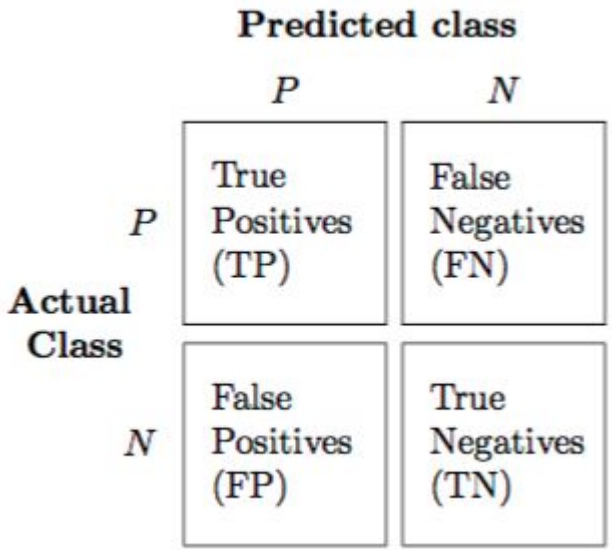
Machine Learning Classifiers

- NAIVE BAYES
- SVM
- kNN
- DECISION TREE
- RANDOM FOREST

Classification Result



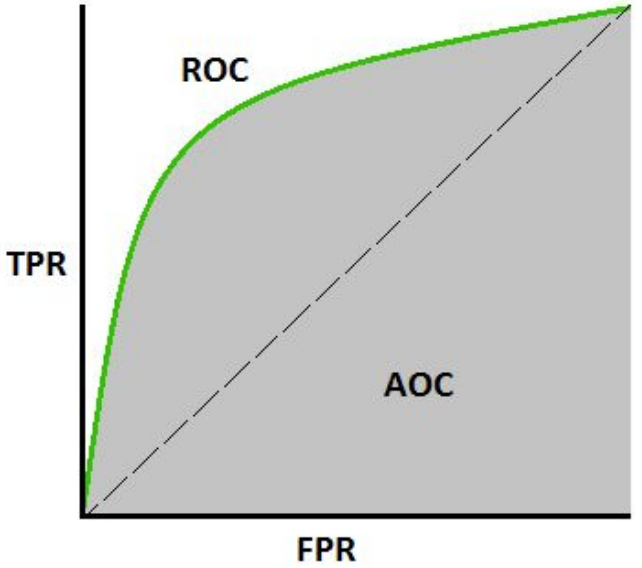
Evaluation and Analysis



Confusion matrix is important to help us understand the performance metrics better

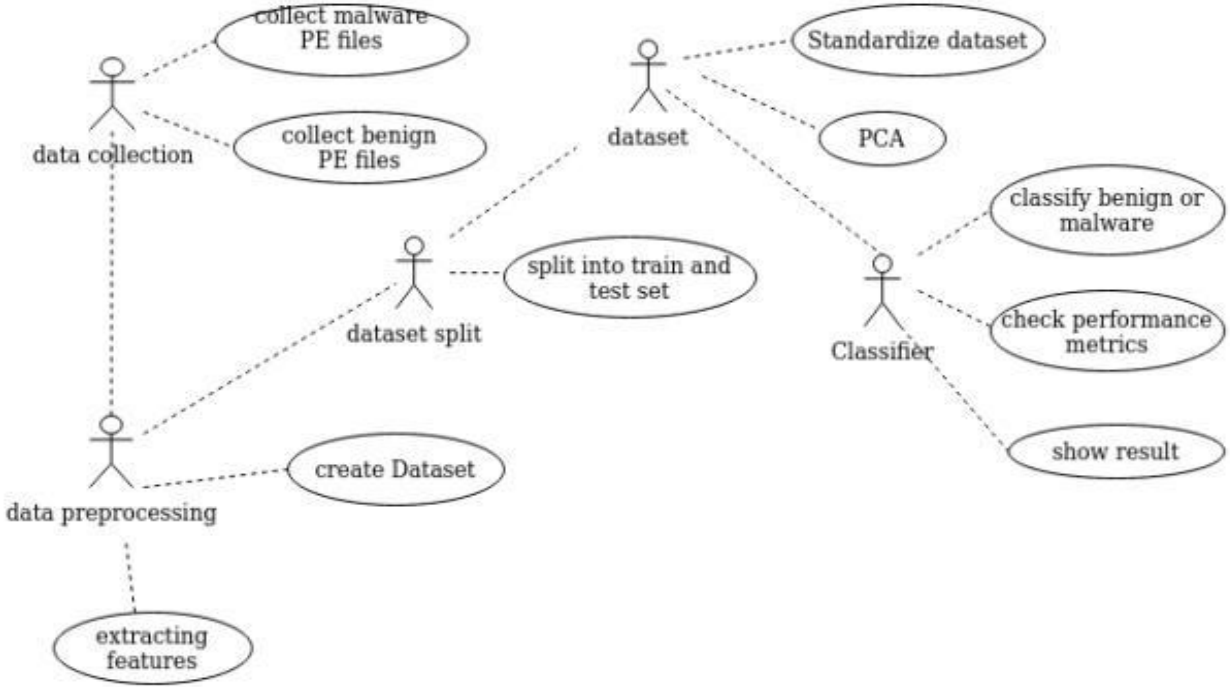
Performance Metrics

Accuracy rate = (TP + TN)/(TP + FP +TN +FN)
Precision = TP/(TP+FP)
Recall = TP/(TP+FN)
F1 score = 2*Precision*Recall/(Precision+Recall)
AOC-ROC curve

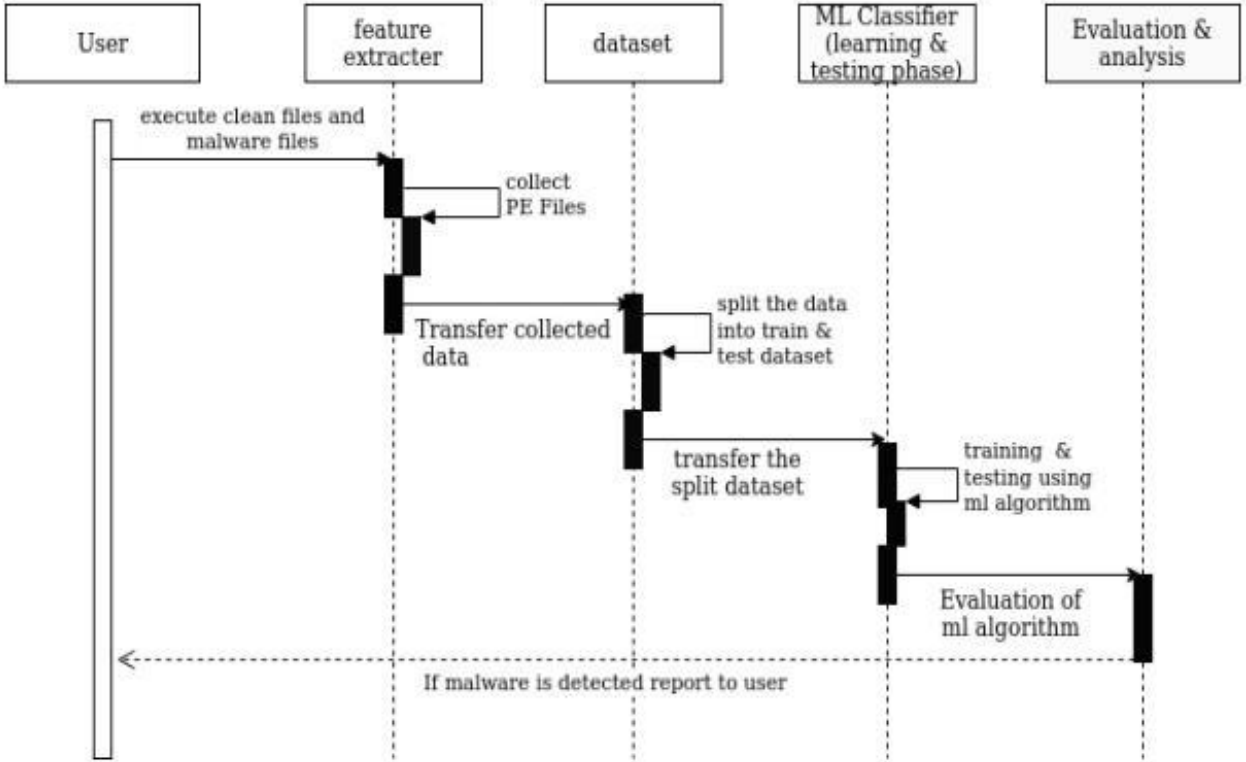


MODELLING and IMPLEMENTATION

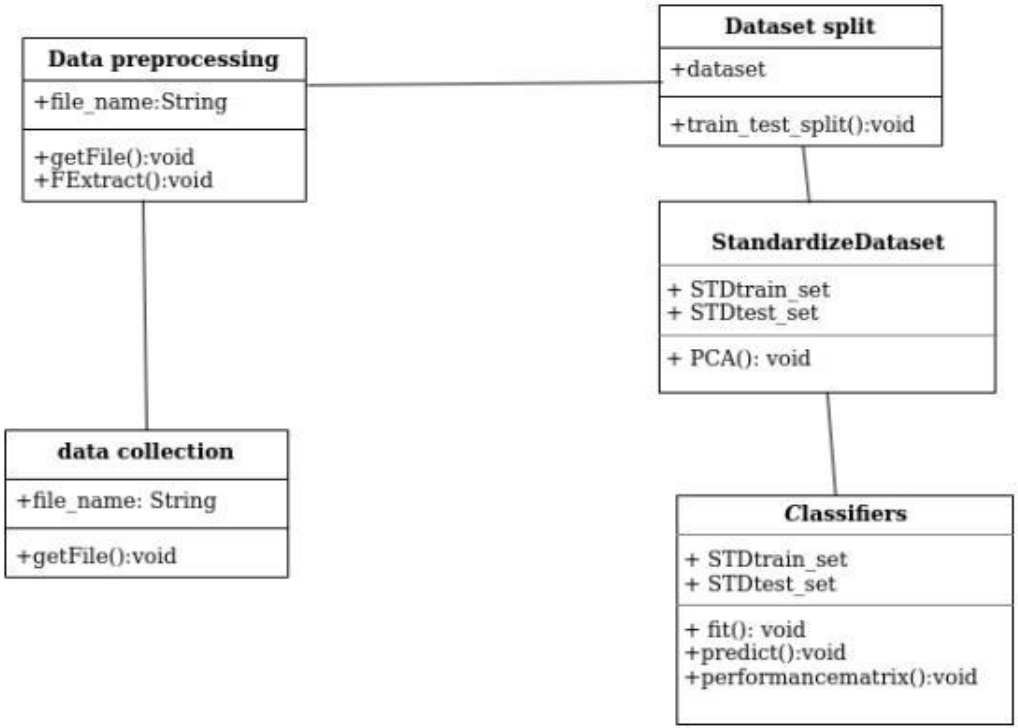
4.1 Use Case Diagram



4.3 Sequence Diagram



4.2 Class Diagram



Tools Used

Jupyter Notebook

Pandas Library

Scikit-learn Library

Matplotlib Library

Implementation Steps

1. Balanced Dataset or not using Shannon's Entropy
2. No variation features dropped
3. Splitting dataset into train or test set
4. Standardization of Dataset
5. Naive Bayes Classifier
6. Knn Classifier
7. Decision Tree Classifier
8. SVM Classifier
9. Random Forest Classifier

RESULTS and ANALYSIS

Accuracy

- Naive Bayes : 49.8%
- kNN : 97.3%
- Decision Tree : 97.4%
- SVM : 96.2%
- Random Forest : 98%

Best Performance : Random Forest

Worst Performance : Naive Bayes

True Positives and False Negatives

True Positive : Malware samples that are correctly classified as malware.

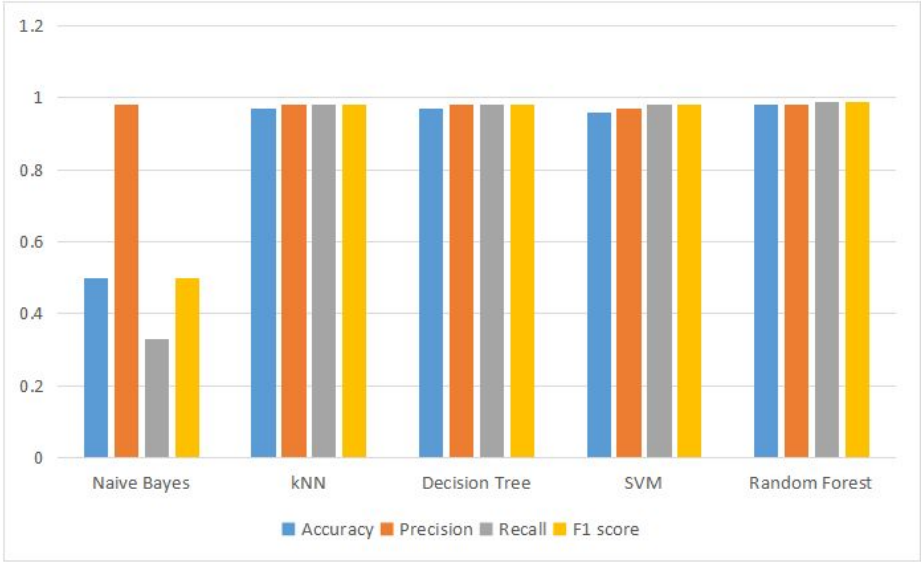
False Negative : Malware samples that are wrongly classified as benign.

Total malware samples: 4386

Classifier	TP	FN
Naive Bayes	1467	2919
kNN	4307	79
Decision Tree	4290	96
SVM	4302	84
Random Forest	4343	43

Classification Results

- It can be clearly seen that Random Forest performed the best among all the classifiers.
- Combines the results of several decision trees.
- Large number of trees make the computation process slow.



THANK YOU