Detection of Malware Attacks Using Machine Learning

PROJECT REPORT

Submitted by

Shivam Gupta, 1854031	Anurag Raj, 1854057	Riya Kundu, 1854073
University Roll No: 12618002048	University Roll No: 12618002019	University Roll No: 12618002042
Registration No: 181260110235	Registration No: 181260110206	Registration No: 181260110229

Under the Supervision of

Prof. Satarupa Biswas

Department of Information Technology

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

KOLKATA, WEST BENGAL

July, 2022

BONAFIDE CERTIFICATE

Certificate that this project report on **“Detection of Malware Attacks Using Machine Learning”** is the bonafide work of **“Shivam Gupta, Anurag Raj and Riya Kundu”**, who carried out the project under my supervision.

SIGNATURE

Prof. (Dr.) Siuli Roy

HEAD OF THE DEPARTMENT

Information Technology

Heritage Institute of Technology.

Kolkata-700107

SIGNATURE

Prof. Satarupa Biswas

PROJECT MENTOR

Information Technology

Heritage Institute of Technology

Kolkata-700107

SIGNATURE

EXTERNAL EXAMNER

ACKNOWLEDGEMENT

We would like to express our sincere thanks and gratitude to our project mentors **Prof. Satarupa Biswas** ma'am and **Prof. Joydev Hazra** sir for letting to work on this project. We are very grateful to them for support and guidance in completing this project. Their guidance and word of encouragement motivated us to achieve our goal.

We would also like to thanks **Prof. Dr. Siuli Roy**, Head of the Department of Information Technology for giving us the opportunity to work on this project.

We are also thankful and fortunate enough to get constant encouragement, support and guidance from all the Teaching and Technical staff of Department of Information technology which helped us in successfully completing our project work.

Shivam Gupta

Anurag Raj

Riya Kundu

ABSTRACT

In today's world the technology is increasing very rapidly and so in the development of malware and malicious activity through which the cyber criminals are gaining a lot of sensitive information and credentials. The innovations of the latest technology motivate cyber criminals to create malicious code and also perform them through which they steal data and perform abnormal activities in the technologies. This is the reason malware detection is very important so that we can detect the malware at the early stage and prevent the devices from getting attacked. There are many detection methods established but the increase of malware that are exploiting the Internet daily has become a serious threat. The manual heuristic inspection of malware analysis is no longer considered effective and efficient compared against the high spreading rate of malware. Hence, automated behavior-based malware detection using machine learning techniques is considered a profound solution. The classifiers used in this research are k-Nearest Neighbors (KNN), Naïve Bayes, J48 Decision Tree, Random Forest, Support Vector Machine (SVM), and Multilayer Perceptron Neural Network (MLP). And also, in this, we will be evaluating the performance based on the algorithm, Detection, and the dataset that will be used, and also confusion matrix will be performed to get the false positive, false negative and recall results.

Keyword: - Malware Detection, Malware Analysis, Classification, SVM, KNN, Decision Tree, Random Forest, MLP, Feature Selection.

TABLE OF CONTENT

TITLE *PAGE NO*

BONAFIDE CERTIFICATE………………………………………………………………………………………. **2**

ACKNOWLEDGEMENT……………………………………………………………………………………. **3**

ABSTRACT…………………………………………………………………………………………………. **4**

CHAPTER 1: INTRODUCTION…………………………………………………………………………. **8**

1.1 Importance of Malware Detection……………………………………………. 9
1.2 Objective………………………………………………………………………………… 10
1.3 Process of implementation of Project………………………………………… 10
1.4 Outline of this Project……………………………………………………………… 11

CHAPTER 2: RELATED WORK AND LITERATURE REVIEW……………………………………. **12**

2.1 Malware Analysis and detection Using Data Mining and Machine Learning Classification………………………………………………………………….. 13
2.2 Malware Analysis and detection Using Data Mining and Machine Learning Classification…………………………………………………………………. 13
2.3 Analysis of Machine Learning Techniques and used in behavior-based Malware Detection……………………………………………………………………. 14

CHAPTER 3: DATA PREPROCESSING AND FEATURE EXTRACTION…………………………… **15**

3.1 Dataset Used………………………………………………………………………… 16
3.2 Data Pre-processing………………………………………………………………… 16
3.3 Feature Engineering………………………………………………………………… 16
3.4 Methods for Feature extraction………………………………………………….. 17
3.4.1 Correlation Metrix with Heat Map………………………………… 17
3.4.2 Univariate Selection……………………………………………………… 17
3.4.3 ExtraTreeClassifier Method…………………………………………… 18
3.4.4 Using Information Gain………………………………………………… 19
3.4.5 Using Xgboost Features importance method………………… 20

CHAPTER 4: HANDING IMBALANCE DATASET……………………………………………………. **22**

4.1 Up sampling Minority Class…………………………………………………… 23

4.2 Down sampling Majority Class…………………………………………………… 24
4.3 Generate Synthetic Data……………………………………………………………... 24
CHAPTER 5: ALGORITHM IMPLEMENTATION…………………………………………………………… 26
5.1 Decision Tree……………………………………………………………………………… 27
5.1.1 Attribute Selection Measures…………………………………………. 27
5.1.2 Advantages and Disadvantages of Decision Tree……………… 28
5.2 Random Forest…………………………………………………………………………… 29
5.2.1 Advantages and Disadvantages of Random Forest…………… 29
5.3 Multi-Layer Perceptron………………………………………………………………… 30
5.3.1 Advantages and Disadvantages of Multi-Layer Perceptron... 30
5.4 KNN (K-Nearest Neighbor) …………………………………………………………… 31
5.4.1 How does KNN works? …………………………………………………. 31
5.4.2 Advantages and Disadvantages of KNN……………………………. 31
5.5 Support Vector Machine (SVM)……………………………………………………. 32
5.5.1 Advantages and Disadvantages of SVM……………………………. 33
5.6 Naïve Bayes………………………………………………………………………………… 33
5.6.1 Bayes Theorem……………………………………………………………… 33
5.6.2 Advantages and Disadvantages of Naïve Bayes…………………. 34
CHAPTER 6: EVALUATION……………………………………………………………………………………… 35
6.1 Confusion Matrix…………………………………………………………………………… 36
6.2 Rate measurement factor……………………………………………………………… 36
6.3 Parameters in Confusion matrix……………………………………………………. 37
CHAPTER 7: EXPERIMENTAL RESULT…………………………………………………………………… 38
7.1 Result of Performance matrix on ExtraTreeClassifier for Unbalance Dataset. 39
7.2 Result of Performance matrix on ExtraTreeClassifier for balance Dataset…… 39
7.3 Result of Performance matrix on Univariate balance Dataset……………………. 40
7.4 Result of Performance matrix on Heatmap balance Dataset……………………. 41
7.5 Result of Performance matrix on Information Gain Balance Dataset………… 42
7.6 Result of Performance matrix on Xgboost Balance Dataset………………………. 42
7.7 Comparative Analysis of Models by Testing on different Dataset for ExtraTreeClassifier and Univariate selection Method……………………………………… 43
CHAPTER 8: CONCLUSION……………………………………………………………………………………… 45
REFERENCES………46

LIST OF FIGURES

TITLE ***PAGE NO***

Figure 1.1 Flow chart of process of implementation of project………………………………………………………………… 10

Figure 3.1 Important features using ExtraTreeClassifier Method…………………………………………………………… 19

Figure 3.2 Important Features using Information Gain……………………………………………………………………….. 20

Figure 3.3 Important Features using Xgboost Method……………………………………………………………………… 21

Figure 4.1 Bar graph showing Imbalance Dataset……………………………………………………………………………... 23

Figure 4.2 Oversampling………. 24

Figure 4.3 Undersampling…….. 24

Figure 4.4 SMOTE……. 25

Figure 6.1 Random Forest Algorithm Tree……………………………………………………………………………………… 29

Figure 6.2 KNN (K-Nearest Neighbor) Algorithm……………………………………………………………………………… 31

Figure 6.3 Support Vector Machine hyperplane……………………………………………………………………………….. 32

Figure 7.1 Result of ExtraTreeClassifier for unbalance dataset……………………………………………………………… 39

Figure 7.2 Result of ExtraTreeClassifier for balance dataset………………………………………………………………… 40

Figure 7.3 Result of Univariate balance dataset………………………………………………………………………………… 41

Figure 7.4 Result of Heatmap balance dataset…………………………………………………………………………………. 41

Figure 7.5 Result of Information Gain for balance dataset………………………………………………………………… 42

Figure 7.6 Result of Xgboost balance dataset………………………………………………………………………………… 43

Figure 7.7 Result of Testing dataset for ExtraTreeClassifier………………………………………………………………….. 43

Figure 7.8 Result of Testing dataset for Univariate…………………………………………………………………………….. 44

CHAPTER 1: INTRODUCTION

Technology and their latest upgrades and updates are increasing day by day and have a very great impact on our day-to-day life. Each and every one is surrounded by technology which is making their life much easier. But this is where cybercriminals have many opportunities to sabotage the new technology by spreading malware in various forms such as through emails, links, documents, etc. This happens because development teams of any IT industry mainly focus on the interface and design of the software instead of securing their network or any method that will prevent from entering of malicious files. But some of the companies are investing or spending billions and billions amount of money to secure their software to the security organization. Also, researchers and development of cybersecurity are working and doing excellent work on establishing new security approaches to prevent cyber-attacks, to explain the term malware, it is nothing but a piece of software that is developed by a cyber attacker in intent to damage a technology device by stealing information, credentials, etc.

1.1 Importance of Malware Detection

Malware is any product tenaciously intended to make harm a PC, server, customer or PC organization. With the fast advancement and development of the web, malware has turned out to be one of the major digital dangers in nowadays. Malware is of different types such as Viruses, Worms, Spywares, Botnet, Ransomware, Trojans, etc. Malware is created by attackers and they only develop it to make money or by any means to gain profit, they also gain money and profit by selling it to the dark web whose bids are the highest, whereas these are the disadvantage of creating malware out the advantage of the malware is that the companies also buy malware to test the security of the software that they developed. Malware on the other hand creates great damage to the reputation of the company and also lots of loss to the company by revealing sensitive information or stealing it, according to McAfee, it is said that the new type of malware is detected every four seconds. it is said that even the antivirus software is not able to detect these types of malwares.

Therefore, to protect computer system from malware is one significant tasks of cybersecurity for a single user as well as for businesses because even a single attack can result in huge information and financial loss.

To mitigate these kinds of problems a better approach for this would be the use of machine learning techniques and their different algorithms because of which the detection rate will much improve to detect malware. The software that will be created will run machine instructions that are created by python and gives us accurate results.

1.2 Objective

The objective of this project is to build a machine learning model which can detect the malware attacks on the dataset. Using different machine learning classification models like SVM, KNN, J48 Decision Tree, Random Forest and Multilayer Perceptron (MLP). And also perform evaluation on machine learning models by confusion Metrix and compares the accuracy, precision value, True Positive rate (TPR), False Positive rate (FPR) and recall value.

This project includes various studies purpose for major development work.

- Study of different Feature extraction methods.
- Study of Data Handing techniques like SMOTE.
- Study of K – Nearest Neighbors (KNN).
- Study of J48 Decision Tree.
- Study of Random Forest Algorithms.
- Study of Malware Detection techniques.
- Study of Evaluation of Machine Learning Models.
- Comparative study of results.

1.3 Process of implementation of Project

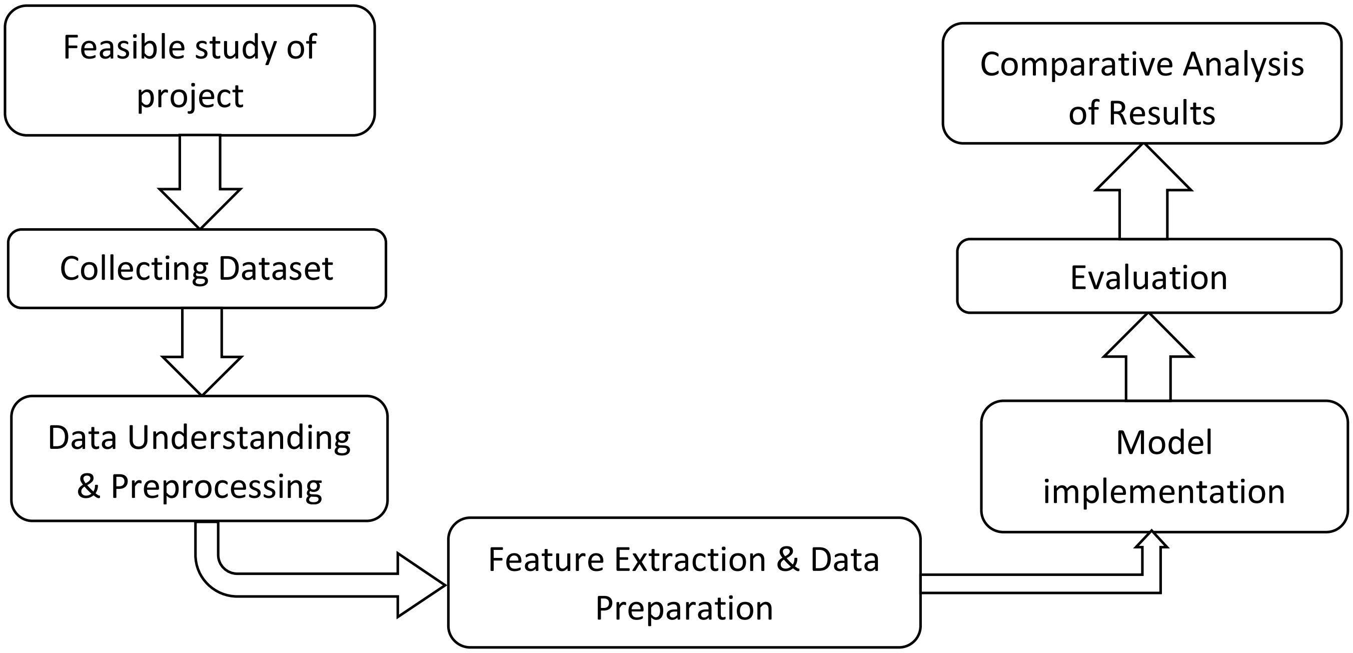
Feasible study of project
Collecting Dataset
Data Understanding & Preprocessing
Feature Extraction & Data Preparation
Model implementation
Evaluation
Comparative Analysis of Results

Figure 1.1 Flow chart of process of implementation of project.

1.4 Outline of this Project

Various studies have been carried out to choose the proper algorithms to study for features selection, selection of the classifiers and the comparative study of results.

The chapters that follow contain the following:

- **Chapter 2:** This chapter contains different research papers and literature review related to this project.
- **Chapter 3:** The chapter describes the data preprocessing and different feature extraction techniques.
- **Chapter 4:** This chapter discuss about handling of imbalance dataset and its techniques.
- **Chapter 5:** This chapter discuss about the machine learning classifiers that we use in our model implementation.
- **Chapter 6:** This chapter holds the evaluation of confusion matrix of the models.
- **Chapter 7:** This chapter discuss about the comparative analysis of the experimental results.
- **Chapter 8:** This chapter bears the conclusion and future scope of the project.

CHAPTER 2: RELATED WORK AND LITERATURE REVIEW

we will be discussing the implementation of malware detection which are proposed by different authors, we will see how they have used different types of classification algorithms and how they have evaluated their models and drawbacks or improvement requirements.

2.1 Malware Analysis and detection Using Data Mining and Machine Learning Classification

Proposed by Sunita Choudhary and Anand Sharma, Mody university of science and technology, Laxmangarh India [6].

- Papers discussed behavioral Based detection methods and advantages from ML algorithms.
- In this paper there are mainly 5 machine learning algorithms are used to classify the Malware.
- KNN, Naïve Bayes, SVM, J48 Decision Tree and MLP (Multi-layer Perceptron).
- Based on Performance Metrics Result the Accuracy rate of SVM is 92.1% and Decision tree is 94.7% with No Feature selection.

With Feature selection the Performance Metrics Result changes little bit, Now the accuracy rate of SVM increase to 93.8% and accuracy rate of decision tree decrease to 92.3%.

2.2 Malware Analysis and detection Using Data Mining and Machine Learning Classification

Proposed by Mozammel Chowdhury, Azizur Rahman, and Rafiqul Islam School of Computing and Mathematics, Charles Sturt University, Wagga Wagga, Australia [7].

- This paper proposes an efficient and robust malware detection scheme using machine learning classification algorithm.
- The dataset is used in this paper contains 52,185 executable files in total consisting of 41,265 malicious files.
- The Machine Learning algorithms used for malware classification is Naïve Bayes, Decision Tree, Random Forest, SVM and a proposed Approach (combines the use of N-gram and API call features).

- The n-gram features are the sequences of substrings with a length of n-bytes extracted from an executable file. API call information shows how malware behaves. API call information shows how malware behaves.
- Based on Performance Metrics Result the accuracy of Proposed Approach is highest 98.6% and SVM accuracy is 95.7% is second best accurate rate.

2.3 Analysis of Machine Learning Techniques and used in behavior-based Malware Detection

Proposed by Ivan Firdausi, Charles Lim, Alva Erwin Department of Information Technology Swiss German University Tangerang, Indonesia [8].

- This paper concluded that a proof of-concept based on automatic behavior-based malware analysis and use of machine learning techniques.
- The data set consists of malware data set and benign instance data set. Both malware and benign instance data sets are in the format of Windows Portable Executable (PE) file binaries.
- A total of 220 unique malware samples were acquired and 250 unique benign software samples were acquired.
- The classifiers used in this research paper are KNN, Naïve Bayes, J48 Decision Tree, SVM and Multi-layer Perceptron Neural Network (MLP).
- Based on the analysis of the tests and experimental results of all the 5 classifiers, the overall best performance achieved by J48 Decision Tree with recall of 95.9% and accuracy of 96.8%.

All the above research papers used different machine learning classification techniques. The research paper 2.2 used a proposed approach combines the use of N-gram and API call features and get highest accuracy of 98.6%.

CHAPTER 3: DATA PREPROCESSING AND FEATURE EXTRACTION

In this step, we collected the dataset of malware and benign application from the website Kaggle. After downloading it we imported it into the excel so that the necessary conversion and data extraction can be performed. Then a combined dataset is prepared to apply it in the different machine learning models.

3.1 Dataset Used

The dataset named “Malware Analysis Datasets: Top-1000 PE Imports Dataset” and The Dataset consist of 45,651 malicious samples and 1929 benign samples. The dataset contains total 47,580 files and 1002 columns.

It contains static analysis data extracted from the 'pe_imports' elements of cuckoo sandbox reports. The dataset almost consisted of everything on the basis of which the detection will be performed. We needed only some columns which we collected there are pre-processing of data.

3.2 Data Pre-processing

The process of Pre-processing of data is very important in machine learning and that is needed to the model that we are using and also it will be necessary to compare it with the other models that we have implemented and compared it to the other model, The feature engineering for our implementation was very easy. The steps taken to do the feature engineering is as explained in detail in the section below.

3.3 Feature Engineering

The Dataset that we created after the preprocessing needs to be transformed into a feature matrix so that we can apply it to our model and the rest of the models that we have compared in machine learning. Feature selection is performed because we can get better accuracy and the performance of detection will get better of the malware and benign files. Further, we labeled the dataset and kept it ready, so the next step is to implement the data into the model that we selected and run it.

Feature selection is nothing but a selection of required independent features. Selecting the important independent features which have more relation with the dependent feature will help to build a good model. The feature matrix of our dataset consists of 0 and 1.

3.4 Methods for Feature extraction

1. Correlation Metrics with Heat Map
2. Univariate Selection
3. ExtraTreeClassifier Method
4. Using Information Gain
5. Using Xgboost Features importance method

3.4.1 Correlation Metrix with Heat Map

Correlation is a term used to represent the statistical measure of linear relationship between two variables. If there are multiple variables and the goal is to find correlation between all of these variables and store them using appropriate data structure, the matrix data structure is used. Such matrix is called as correlation matrix.

It will give the relation between dependent and independent features. Heatmap is a graphical representation of 2D (two-dimensional) data. Each data value represents in a matrix. The value of correlation can take any values from -1 to 1.

Formula for Pearson correlation or correlation (r):

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

r = correlation coefficient

$x_i =$ values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

3.4.2 Univariate Selection

In this, Statistical tests (like chi-square, F-score) can be used to select the independent features which have the strongest relationship with the dependent feature. Here we use the chi-squared (chi²) statistical test for non-negative features to select the best features. Which feature has the highest feature score will be more related to the dependent feature and choose those features for the model. Cut-off feature score is determined by user.

We select top 107 features which are most important based on feature score using univariate selection method for model building.

Formula for Chi-Square test: -

$$x^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

x^2 = chi squared

O_i = observed value

E_i = expected value

3.4.3 ExtraTreeClassifier Method

ExtraTreesClassifier is **an ensemble learning method fundamentally based on decision trees**. ExtraTreesClassifier, like Random Forest, randomizes certain decisions and subsets of data to minimize over-learning from the data and overfitting.

In this method, the ExtraTreesClassifier method will help to give the importance of each independent feature with a dependent feature. Feature importance will give you a score for each feature of your data, the higher the score more important or relevant to the feature towards your output variable.

We select most important 94 features out 1002 features based on feature score of ExtraTreeClassifier Method.

Some Important Features by using this method: -

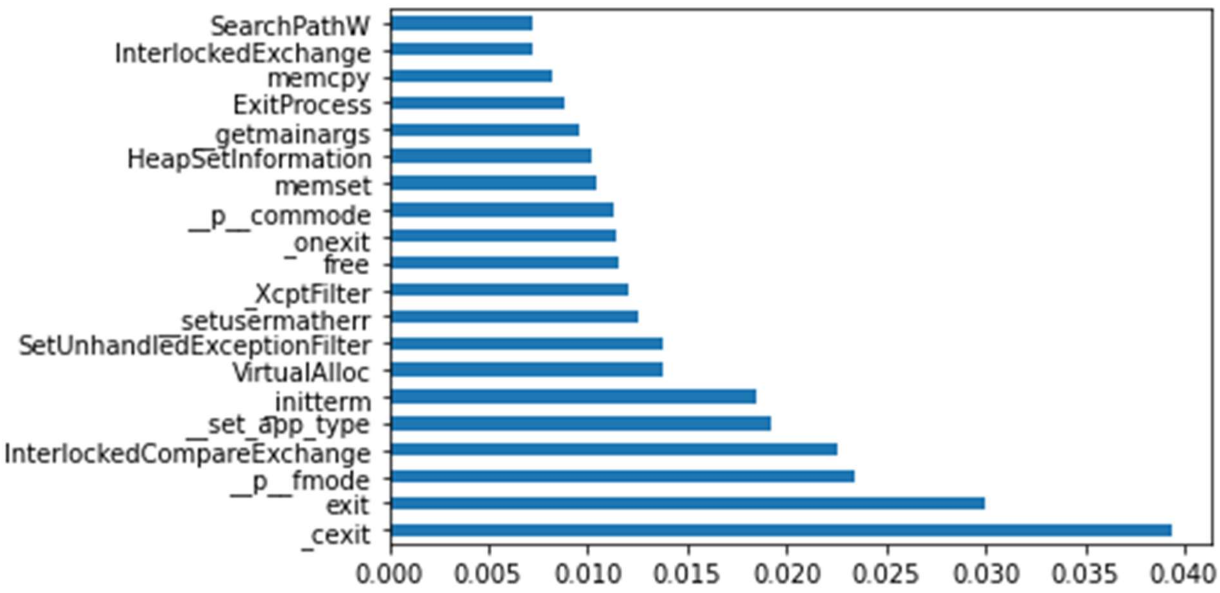

Figure 3.1 Important features using ExtraTreeClassifier Method

3.4.4 Using Information Gain

Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.

We select 113 important features out of 1002 based on good information gain value.

Formula for Information Gain (IG)

Entropy is defined as

$$H(X) = -\sum_i p_i \, log_2(p_i)$$

Where, i is the number of different values that X can take.

Average specific conditional Entropy of X is gives as

$$H\left(\frac{Y}{X}\right) = \sum_i P\,(X = v)\,H(\frac{Y}{X=v})$$

Information Gain (IG): - $IG(Y/X) = H(Y) - H(Y/X)$

Some Important Features by using this Method: -

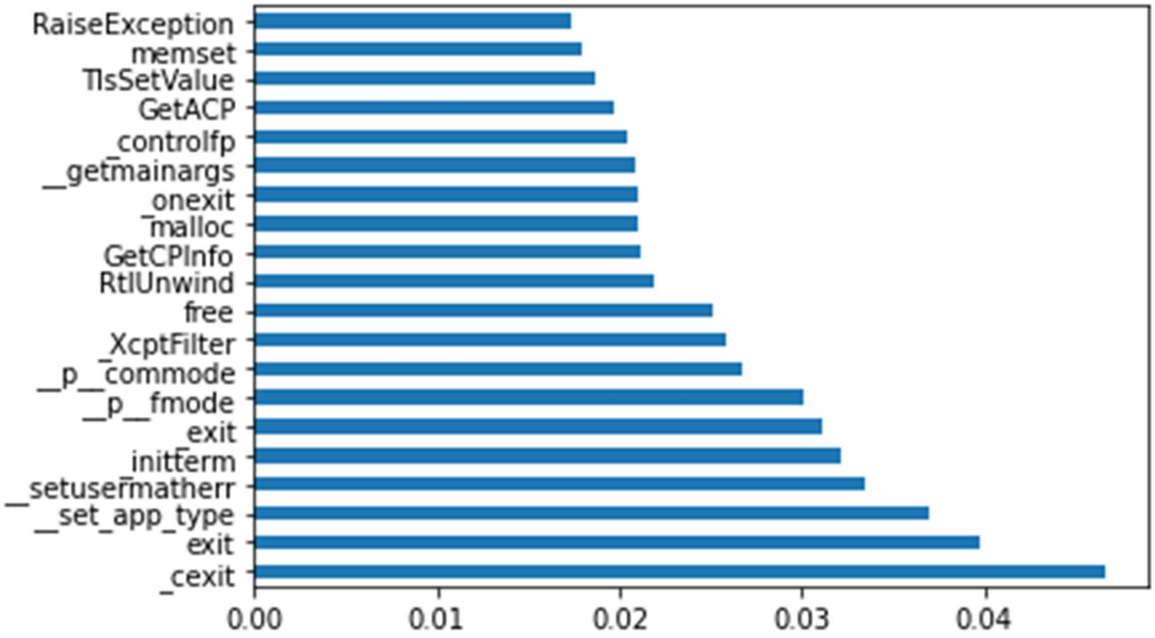

Figure 3.2 Important Features using Information Gain

3.4.5 Using Xgboost Features importance method

You can get the feature importance of each feature of your dataset by using the feature importance property of the model. Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.

We select 107 important features out of 1002 based on Xgboost importance features score.

Some Important Features by using Xgboost: -

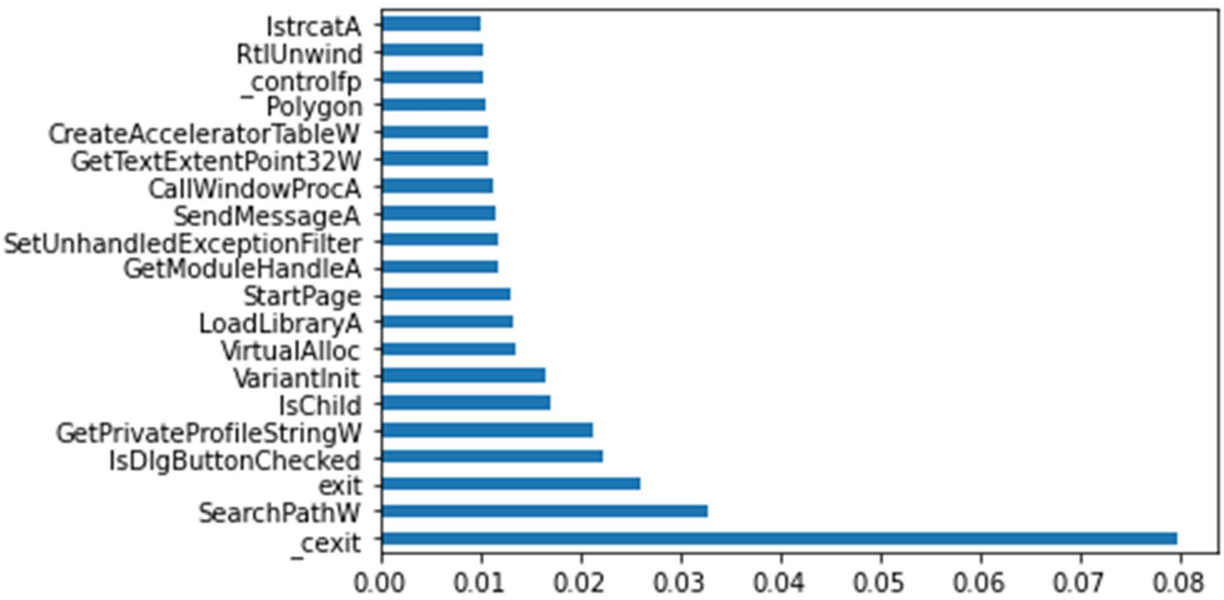

Figure 3.3 Important Features using Xgboost Method

CHAPTER 4: HANDING IMBALANCE DATASET

After data pre-processing, we observe our original dataset is imbalance and imbalance dataset give incorrect result and imbalance dataset has high false positive rate and that is not good for our models. Imbalanced dataset is quite common problem of classification.

Imbalanced data refers to those types of datasets where the target class has an uneven distribution of observations, i.e., one class label has a very high number of observations and the other has a very low number of observations.

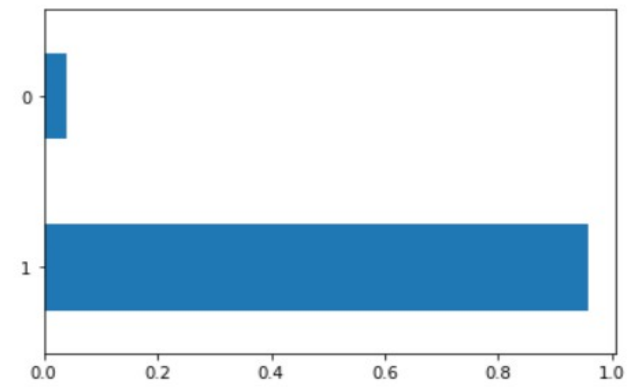

Figure 4.1 Bar graph showing Imbalance Dataset

1's represents malicious files and 0's benign files. Here we can clearly see the malicious files are very high than benign files. And that is called imbalance data. False positive rate of models is high if dataset is imbalance means predicting the result is positive(malicious) but actually the result is negative(benign).

Hence, handling the imbalance in the dataset is essential prior to model training.

There are various methods to balance the dataset like: -

1. Up sampling Minority Class (Oversampling)
2. Down sampling Majority Class (Undersampling)
3. Generate Synthetic Data (SMOTE)

4.1 Up sampling Minority Class

Up sampling or Oversampling refers to the technique to create artificial or duplicate data points or of the minority class sample to balance the class label.

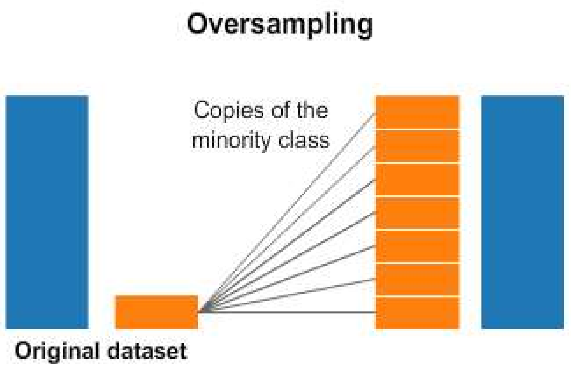

Figure 4.2 Oversampling [1]

4.2 Down sampling Majority Class

Down sampling or Undersampling refers to remove or reduce the majority of class samples to balance the class label.

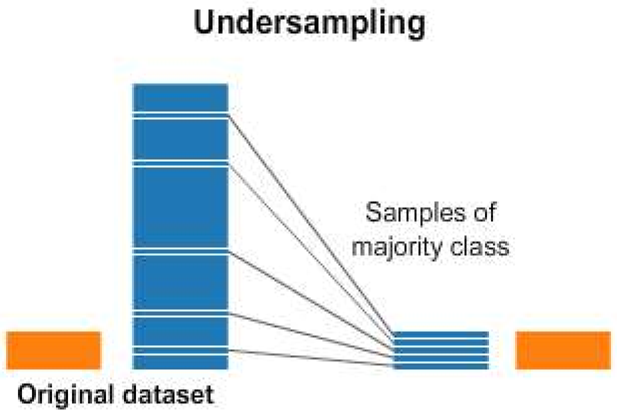

Figure 4.3 Undersampling [2]

4.3 Generate Synthetic Data

Undersampling techniques are not recommended as it removes the majority class data points. Generating synthetic data points of minority samples is a type of oversampling technique. The idea is to generate synthetic data points of minority class samples in the nearby region or neighborhood of minority class samples.

SMOTE (Synthetic Minority Over-Sampling Techniques) is a popular synthetic date generation oversampling technique that utilizes the **K-nearest neighbor (KNN)** algorithm to create synthetic data.

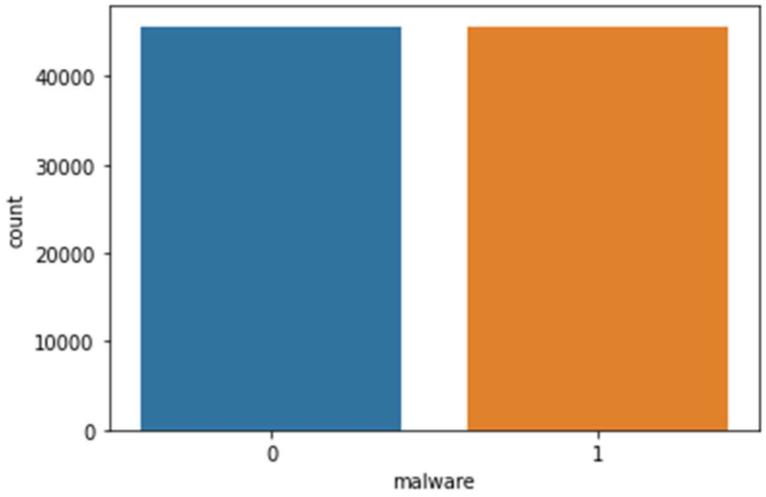

Figure 4.4 SMOTE

SMOTE is one of best method for balancing Data. That’s why we used SMOTE Technique to balance our dataset.

CHAPTER 5: ALGORITHM IMPLEMENTATION

In the implementation step, the dataset that we created was implemented in each machine learning algorithm. Further moving on to the Implementation or applying model we split the dataset into two parts that are 70% of the data is aligned to training and 30% of data is allotted to test data, this is done because we can test the performance of the data instead of applying the whole data to training. We did the split by applying train_test_split which is imported from sklearn model package in python. Explanation of each model which is the KNN, SVM Model, Decision Tree Model, Random Forest with Decision Tree, Naïve Bayes and Multi-Layer Perceptron (MLP) model is explained in details below.

5.1 Decision Tree

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data.

Important Terminology related to Decision Trees [9]: -

- **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.
- **Leaf Node:** Nodes do not split is called Leaf Node.
- **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- **Branch:** A Sub-section of the entire tree is called branch.
- **Parent and child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

5.1.1 Attribute Selection Measures

If the dataset consists of N attributes, then deciding which attribute to place at the root or at different levels of the trees as internal nodes is complicated step. By just randomly selecting any node to be the root can’t solve the issue.

For solving this attribute selection problems, researchers worked and devised some solutions. They suggested using some criteria like: -

- **Entropy:** Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information.
- **Information Gain:** Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification. Information gain is a decrease in entropy.
- **Gini Index:** You can understand the Gini index as a cost function used to evaluate splits in the dataset. It is calculated by subtracting the sum of squared probabilities of each class from one.

$$\boldsymbol{Gini = 1 - \sum_{i=1}^{c} (p_i)^2}$$

- **Gain ratio:** Information gain is biased towards choosing attributes with a large number of values as root nodes.

$$\text{Gini} = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy(before) - \sum_{j=1}^{K} Entropy(j, after)}{\sum_{j=1}^{K} w_j \ log_2 \ w_j}$$

- **Chi-Square:** The acronym CHAID stands for chi-squared Automatic Interaction Detector. It is one of the oldest tree classification methods. It finds out the statistical significance between the differences between sub-nodes and parent node.

5.1.2 Advantages and Disadvantages of Decision Tree: -

- **Advantages: -**
  - ➢ Decision trees are easy to interpret and visualize.
  - ➢ It can be used for feature engineering such as predicting missing values, suitable for variable selection.
  - ➢ Compared to other algorithms decision trees requires less efforts for data preparation during pre-processing.
  - ➢ A decision tree does not require scaling of data as well.
- **Disadvantages: -**

- A small change in the data can cause large change in the structure of the decision tree causing instability.
- For a Decision tree sometimes, calculation can go far more complex compared to other algorithms.
- Decision trees may become very large and complex with large number of attributes.

5.2 Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

The below diagram explains the working of the Random Forest algorithm: -

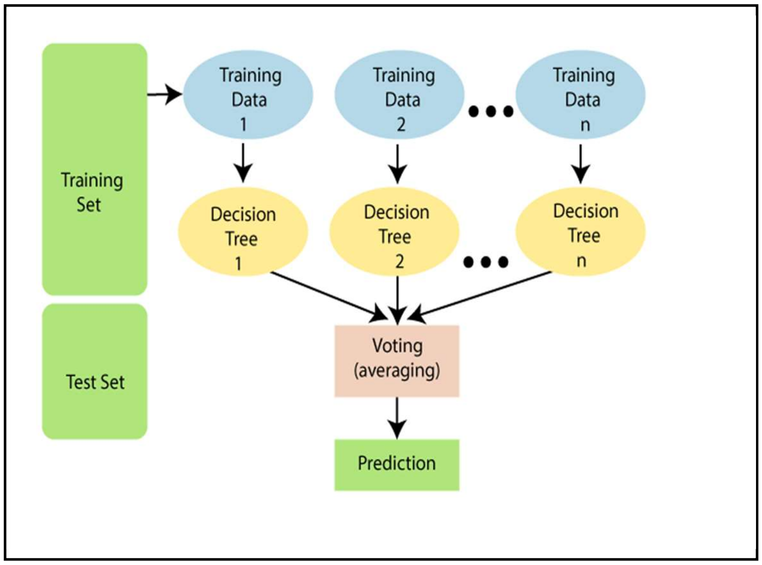

Figure 6.1 Random Forest Algorithm Tree [3]

5.2.1 Advantages and Disadvantages of Random Forest: -

- **Advantages: -**
    - Random forest is capable of performing both classification and Regression tasks.
    - It is capable of handling large datasets with high dimensionality.
    - It enhances the accuracy of the model and prevents the overfitting issue.
- **Disadvantages: -**
    - Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

5.3 Multi-Layer Perceptron

The perceptron is an algorithm for learning a binary classifier called threshold function. A function that maps its input x (a real valued vector) to an output value f(x) (single binary value).

Multi-Layer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers- the input layer, output layer and hidden layer. The input layer receives the input signal to be processed. The required task such as prediction and classification is performed by output layer.

5.3.1 Advantages and Disadvantages of Multi-Layer Perceptron: -

- **Advantages: -**
    - Capability to learn non-linear models.
    - Capability to learn models in real time.
    - The same accuracy ratio can be achieved even with smaller data.
- **Disadvantages: -**
    - MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore, different random weight initializations can lead to different validation accuracy.
    - MLP requires tuning a number of hyperparameters such as the number of hidden neurons, layers and iterations.
    - MLP is sensitive to feature scaling.

5.4 KNN (K-Nearest Neighbor)

K Nearest Neighbor (KNN) is intuitive to understand and easy to implement the algorithm. Beginners can master this algorithm even in the early phases of their Machine Learning studies.

5.4.1 HOW DOES KNN WORKS?

Consider the following figure. Let us say we have plotted data points from our training set on a two-dimensional feature space. As shown, we have a total 10 data points (5 yellow and 5 blue). Yellow data points belong to class1 and blue data points belong to class2. And star data point in a feature space represents the new point for which a class is to be predicted. Obviously, we say it belongs to class (yellow points) [10].

Why?

Because its nearest neighbors belong to that class!

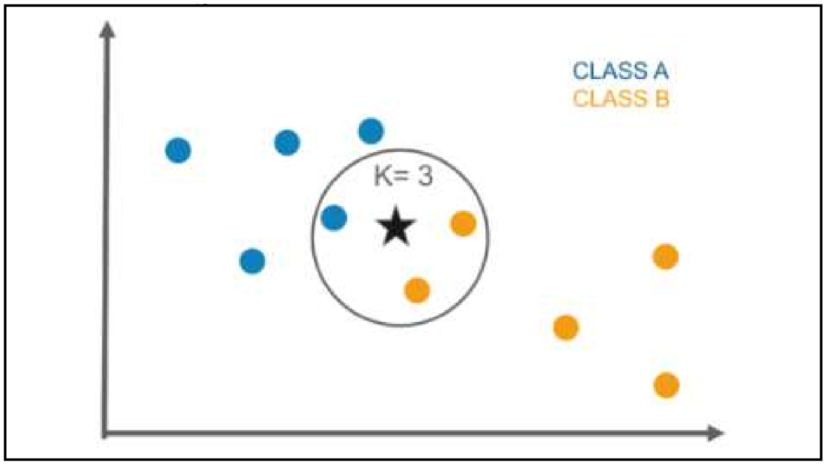

Figure 6.2 KNN (K-Nearest Neighbor) Algorithm [4]

5.4.2 Advantages and Disadvantages of KNN (K- Nearest Neighbor): -

- **Advantages: -**
  - ➢ No training periods. KNN is lazy learner. It does not learn anything in training period.
  - ➢ KNN is very easy to implement, there are only two parameters required to implement KNN I.e., the value of K and distance function.

- **Disadvantages: -**
  - Does not work well with large dataset.
  - Need feature scaling. We need to do feature scaling before applying KNN algorithm to any dataset, if we don’t do so, KNN may generate wrong predictions.
  - Sensitivity to noisy data, missing values outliers.

5.5 SVM (SUPPORT VECTOR MACHINE)

Support vector machine or SVM is one of the most popular supervised learning algorithms, which is used for classification as well as Regression problems. However, primarily, it is used for classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can the correct category in the future. The best decision boundary is called hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

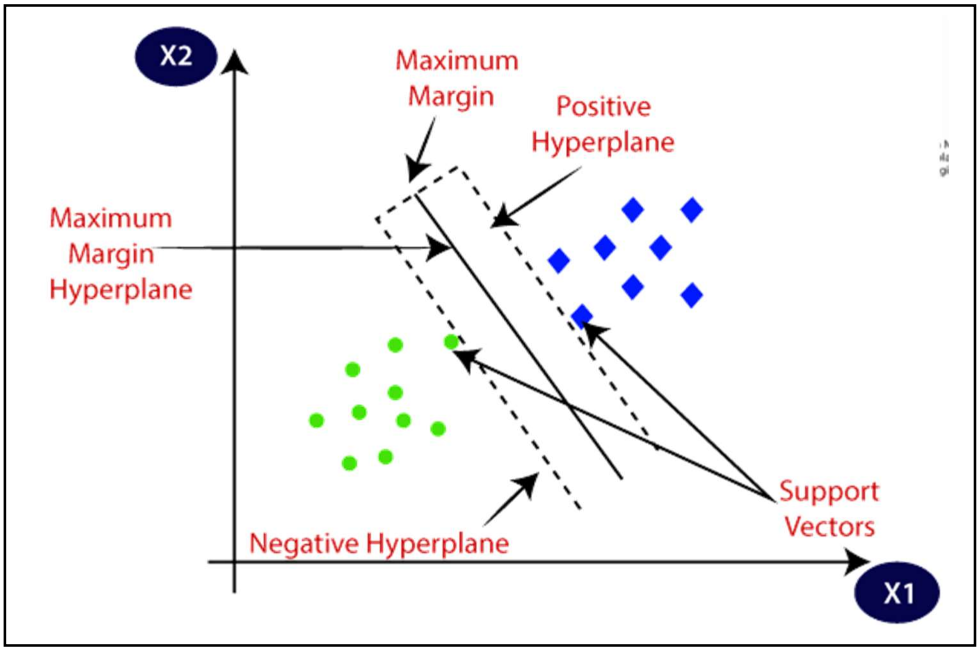

Figure 6.3 Support Vector Machine hyperplane [5]

5.5.1 Advantages and Disadvantages of SVM (Support vector machine): -

- **Advantages: -**
  - It works really well with a clear margin of separation.
  - It is effective in high dimensional spaces.
  - It is effective in cases where the number of dimensions is greater than the number of samples.
- **Disadvantages: -**
  - It doesn't perform well when we have large dataset because the required training time is higher.
  - It also doesn't perform well when dataset has more noise i.e. target classes are overlapping.

5.6 NAÏVE BAYES

Naïve Bayes classifiers are a collection of classification algorithms based on Bayes Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle i.e., every pair of features being classified is independent of each other.

5.6.1 Bayes Theorem

Bayes theorem is also known as Bayes Rule or Bayes Rule or Bayes law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on Conditional probability.

The formulae for Bayes theorem are given as: -

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where

P (A|B) is Posterior probability of hypothesis A on the observed event B.

P (B|A) is Likelihood probability. Probability of the evidence given that the probability of a hypothesis is true.

P (A) is Prior Probability. Probability of hypothesis before observing the evidence.

P (B) is Marginal Probability. Probability of Evidence.

5.6.2 Advantages and Disadvantages of Naïve Bayes: -

- **Advantages: -**
  - It is easy and fast to predict class of test data set.
  - It performs well in case of categorical input variables compared to numeric variables.
- **Disadvantages: -**
  - If categorical variable has a category, which was not observed in training data set, then model will assign a 0 probability and unable to make prediction.
  - Another limitation of Naïve Bayes is the assumption of independent predictors.
  - In real life, it is almost impossible that we get a set of predictors which are completely independent.

CHAPTER 6: EVALUATION

A confusion matrix is an N * N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making [11].

6.1 Confusion Matrix

For binary classification problem, we would have a 2 * 2 matrix as shown below with 4 values.

Values	**Predicted: Negative (0)**	**Predicted: Positive (1)**
Actual: Negative (0)	**TN**	**FP**
Actual: Positive (1)	**FN**	**TP**

- **True Positive (TP):**
  - The predicted value matches the actual value.
  - The actual value was positive and the model predicted a positive value.
- **True Negative (TN):**
  - The predicted value matches the actual value.
  - The actual value was negative and model predicted a negative value.
- **False Positive (FP):**
  - The predicted value was falsely predicted.
  - The actual value was negative but the model predicted a positive value.
- **False Negative (FN):**
  - The predicted value was falsely predicted.
  - The actual value was positive but the model predicted a negative value.

6.2 Rate Measurement Factor

Rate is a measure factor in confusion matrix. It has also 4 types TPR, FPR, TNR, FNR.

- True Positive Rate (TPR): **TP / (TP + FN)**
- False Positive Rate (FPR): **FP / (TN + FP)**
- False Negative Rate (FNR): **FN / (TP + FN)**

- True Negative Rate (TNR): **TN / (TN + FP)**

6.3 Parameters in Confusion Matrix

The parameters that are considered in the confusion matrix such as Accuracy, Precision, recall, are explained below and also the equation that is used to calculate those.

- **Precision:** A metric that quantifies the number of correct positive detection made. This calculates the accuracy of the minority class. It is calculated by correctly detected positives divided by the total number of positives detected.

Precision = True Positives / (True Positives + False Positives)

- **Accuracy:** Accuracy means the detected values by model which are correctly and as told to the model to do. It is calculated by adding TP and TN divided by all the positives and negatives.

Accuracy = TP + TN / TP + TN + FP + FN

- **F1-Score:** F1-Score is a metric in which precision and recall are considered. The equation to calculate is below:

F1-Score = 2 * (Precision * Recall / (Precision + Recall))

- **Recall:** Recall is the sensitivity or the True positives rate of which the equation is explained below.

Recall = True Positives / (True Positives + False Negatives)

- **Cross-Validation: -**

Cross-Validation is a technique for validating the model efficiency by training on the subset of input data and testing on previously unseen subset of the input data.

CHAPTER 7: EXPERIMENTAL RESULT

7.1 Result of Performance matrix on ExtraTreeClassifier (For Unbalance Dataset)

ML Models	*True Positive rate (TPR)*	*False Positive rate (FPR)*	*Precision Value*	*Recall Value*	*Overall Accuracy*
KNN	99.57%	41.08%	98.34%	99.57%	97.98%
SVM	99.65%	55.73%	97.81%	99.65%	97.51%
Naïve Bayes	94.81%	30.02%	98.51%	94.81%	93.68%
Decision Tree	99.14%	39.89%	98.41%	99.14%	97.64%
Random Forest	99.59%	40.16%	98.41%	99.59%	98.06%
Multi-Layer Perceptron (MLP)	99.37%	41.53%	98.35%	99.37%	97.80%

Figure 7.1 Result of ExtraTreeClassifier for unbalance dataset

Based on the above Figure 7.1, Overall Accuracy of Random Forest classifier is highest 98.06% among all of them and Naïve Bayes gives lowest accuracy 93.68% for the ExtraTreeClassifier unbalance dataset. We can see the false positive rate (FPR) is higher than expected because of the dataset is unbalance and unbalance dataset cause a false result. The precision value of random forest is 98.41% and the recall value is also highest for random forest classifier is 99.59%. Imbalanced classification is the problem of classification when there is an unequal distribution of classes in the training dataset.

7.2 Result of Performance matrix on ExtraTreeClassifier (For Balance Dataset)

ML Models	*True Positive rate (TPR)*	*False Positive rate (FPR)*	*Precision Value*	*Recall Value*	*Overall Accuracy*
KNN	95.59%	3.16%	96.79%	95.59%	96.21%

SVM	91.71%	8.55%	91.49%	91.71%	91.57%
Naïve Bayes	95.33%	32.33%	74.73%	95.33%	81.52%
Decision Tree	96.53%	3.01%	96.97%	96.53%	96.76%
Random Forest	96.93%	2.92%	97.07%	96.93%	97.00%
Multi-Layer Perceptron	96.07%	3.65%	96.34%	96.07%	96.21%

Figure 7.2 Result of ExtraTreeClassifier for balance dataset

-

Based on the above Figure 7.2, Overall Accuracy of Random Forest classifier is highest 97.00% among all of them and Naïve Bayes gives lowest accuracy 81.52% for the ExtraTreeClassifier balance dataset. We can see the false positive rate (FPR) is lesser because of the dataset is balanced. The precision value of random forest is 96.34% and the recall value is also highest for random forest classifier is 96.07%.

Here we notice that having a balanced data set for a model would generate higher accuracy models, higher balanced accuracy and balanced detection rate and less false positive rate (FPR).

7.3 Result of Performance matrix on Univariate Balance Dataset

ML Models	*True Positive rate (TPR)*	*False Positive rate (FPR)*	*Precision Value*	*Recall Value*	*Overall Accuracy*
KNN	95.20%	13.99%	87.17%	95.20%	90.60%
SVM	94.39%	19.56%	82.88%	94.39%	87.42%
Naïve Bayes	87.95%	29.54%	74.91%	87.95%	79.21%
Decision Tree	96.59%	14.29%	87.09%	96.59%	91.14%
Random Forest	96.91%	14.19%	87.22%	96.91%	91.35%

Multi-Layer Perceptron (MLP)	96.19%	14.10%	87.25%	96.19%	91.05%

Figure 7.3 Result of Univariate balance dataset

Based on the above Figure 7.3, Overall Accuracy of Random Forest classifier is highest 91.35% among all of them and Naïve Bayes gives lowest accuracy 79.21% for the Univariate balance dataset. The precision value of random forest is 87.22% and the recall value is also highest for random forest classifier is 96.91%.

7.4 Result of Performance matrix on Heatmap Balance Dataset

ML Models	*True Positive Rate (TPR)*	*False Positive rate (FPR)*	*Precision Value*	*Recall Value*	*Overall Accuracy*
KNN	88.45%	2.35%	97.40%	88.45%	93.05%
SVM	90.56%	7.83%	92.06%	90.56%	91.36%
Naïve Bayes	36.49%	3.08%	92.23%	36.49%	66.65%
Decision Tree	95.75%	3.85%	96.12%	95.75%	95.94%
Random Forest	96.20%	3.80%	96.19%	96.20%	96.20%
Multi-Layer Perceptron (MLP)	95.33%	4.09%	95.89%	95.33%	95.61%

Figure 7.4 Result of Heatmap balance dataset

Based on the above Figure 7.4, Overall Accuracy of Random Forest classifier is highest 96.20% among all of them and Naïve Bayes gives lowest accuracy 66.65% for the Heatmap balance dataset. The precision value of random forest is 96.19% and the recall value is also highest for random forest classifier is 96.20%.

7.5 Result of Performance matrix on Information Gain Balance Dataset

ML Models	True Positive rate (TPR)	False Positive rate (FPR)	Precision Value	Recall Value	Overall Accuracy
KNN	95.50%	6.17%	93.92%	95.50%	94.66%
SVM	91.93%	9.50%	90.66%	91.93%	91.22%
Naïve Bayes	68.22%	21.41%	76.17%	68.22%	73.39%
Decision Tree	95.79%	4.25%	95.75%	95.79%	95.77%
Random Forest	96.01%	4.16%	95.84%	96.01%	95.92%
Multi-Layer Perceptron	95.12%	4.26%	95.73%	95.125	95.13%

Figure 7.5 Result of Information Gain for balance dataset

Based on the above Figure 7.5, Overall Accuracy of Random Forest classifier is highest 95.92% among all of them and Naïve Bayes gives lowest accuracy 73.39% for the Information gain balance dataset. The precision value of random forest is 95.84% and the recall value is also highest for random forest classifier is 96.01%.

7.6 Result of Performance matrix on Xgboost Balance Dataset

ML Models	True Positive rate (TPR)	False Positive rate (FPR)	Precision Value	Recall Value	Overall Accuracy
KNN	94.79%	4.01%	95.93%	94.79%	95.39%
SVM	92.47%	7.28%	92.72%	92.47%	92.59%
Naïve Bayes	73.94%	12.92%	85.17%	73.94%	80.49%

Decision Tree	96.42%	4.58%	95.46%	96.42%	95.92%
Random Forest	96.85%	4.54%	99.51%	96.85%	96.15%
Multi-Layer Perceptron (MLP)	94.13%	3.23%	96.67%	94.13%	95.44%

Figure 7.6 Result of Xgboost balance dataset

Based on the above Figure 7.6, Overall Accuracy of Random Forest classifier is highest 96.15% among all of them and Naïve Bayes gives lowest accuracy 80.49% for the Information gain balance dataset. The precision value of random forest is 99.51% and the recall value is also highest for random forest classifier is 96.85%.

7.7 Comparative Analysis of Models by Testing on different Dataset for Extra Tree Classifier Method and Univariate selection method.

ML Models	*True Positive rate (TPR)*	*False Positive rate (FPR)*	*Precision Value*	*Recall Value*	*Overall Accuracy*
KNN	96.00%	2.88%	96.91%	96.00%	96.57%
SVM	91.66%	8.20%	91.34%	91.66%	91.73%
Naïve Bayes	95.49%	31.98%	73.81%	95.49%	81.35%
Decision Tree	96.98%	2.73%	97.13%	96.98%	97.14%
Random Forest	97.07%	2.70%	97.13%	97.07%	97.18%
Multi-Layer Perceptron (MLP)	96.64%	3.32%	96.48%	96.64%	96.61%

Figure 7.7 Result of Testing dataset for ExtraTreeClassifier

ML Models	True Positive rate (TPR)	False Positive rate (FPR)	Precision Value	Recall Value	Overall Accuracy
KNN	95.73%	13.91%	86.26%	95.73%	90.68%
SVM	94.19%	20.02%	82.42%	94.19%	86.49%
Naïve Bayes	87.51%	30.26%	72.52%	87.51%	78.21%
Decision Tree	97.24%	14.04%	86.34%	97.24%	91.34%
Random Forest	97.31%	14.02%	86.36%	97.31%	91.38%
Multi-Layer Perceptron (MLP)	97.21%	14.50%	85.95%	97.21%	91.08%

Figure 7.8 Result of Testing dataset for Univariate

Figure 7.7 and Figure 7.8 showing the result of all the models testing on the completely new dataset. We only test on two feature extraction methods ExtraTreeClassifier and Univariate selection Method. Random Forest classifier accuracy is highest and Naïve bayes accuracy is lowest among all the classifiers output.

CHAPTER 8: CONCLUSION

We have also discussed about different machine learning algorithms that can be of great help in detecting malware as with the quick development and advancement of web, malware is major threat.

Data Balancing is very important step for the classification Problems because unbalance data may give the false results and that is not good for our model. We use SMOTE data balancing techniques to balance our dataset.

As we have implemented five features engineering methods on six different machine learning algorithms.

The following results we got: -

In ExtraTreeClassifier the Random Forest Accuracy is 98.06%

In Univariate Feature selection method, the Random Forest Accuracy is 91.35%

In Heat Map or Correlation Matrix method, the Random Forest Accuracy is 96.20%

In Information Gain method, the Random Forest Accuracy is 95.92%

In Xgboost Feature importance method, the Random Forest Accuracy is 96.15%

The overall best performance was achieved by Random Forest with Feature selection on balanced ExtraTreeClassifier Dataset, with a True Positive Rate (TPR) of 96.93 %, False positive Rate (FPR) of 2.92% and the accuracy of 97.00 %. Based on experimental result we came at conclusion the ExtraTreeClassifier Feature selection gives best result on both experimental result and Testing result and the Univariate Selection method gives comparatively lower result in all the models.

Based on the result we came to the conclusion The Random Forest Algorithm performs better than other machine learning algorithms among all the feature engineering dataset.

References

[1] https://towardsdatascience.com/5-techniques-to-work-with-imbalanced-data-in-machine-learning-80836d45d30c

[2] https://towardsdatascience.com/5-techniques-to-work-with-imbalanced-data-in-machine-learning-80836d45d30c

[3] https://www.javatpoint.com/machine-learning-random-forest-algorithm

[4] https://www.edureka.co/blog/k-nearest-neighbors-algorithm

[5] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[6] Malware Detection & Classification using Machine Learning Proposed by Sunita Choudhary Computer Science and Engineering Mody University of Science and Technology Laxmangarh, India and Anand Sharma Computer Science and Engineering Mody University of Science and Technology.

[7] Malware Analysis and Detection Using Data Mining and Machine Learning Classification by Mozammel Chowdhury, Azizur Rahman, and Rafiqul Islam School of Computing and Mathematics, Charles Sturt University, Wagga Wagga, Australia.

[8] Analysis of machine learning techniques used in behavior-based malware detection by Ivan Firdausi, Charles Lim, Alva Erwin Department of Information Technology Swiss German University Tangerang, Indonesia.

[9] https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[10] https://www.analyticsvidhya.com/blog/2021/04/simple-understanding-and-implementation-of-knn-algorithm/

[11] https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/