MALLA REDDY UNIVERSITY
(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI & ML)

MALWARE DETECTION BASED ON BEHAVIOUR MODEL

Design & Developed by

A. SRI CHARAN	2011CS020019
A. SURYAPAVAN	2011CS020020
A. V. S. PAVAN	2011CS020021
A. VIJAY	2011CS020022

GUIDED BY

Dr. R.V.S.S.S.Nagini

Department of Computer Science & Engineering (AI&ML)

MALLA REDDY UNIVERSITY

2020-2024

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

COLLEGE CERTIFICATE

This is to certify that this is the bonafide record of the application development entitled, ”MALWARE DETECTION BASED ON BEHAVIOUR MODEL” Submitted by A .Sri charan (2011CS020019) , A. Surya pavan (2011CS020020) , A.V.S.Pavan(2011CS020021), A. Vijay (2011CS020022) B.Tech III year I semester, Department of CSE (AI&ML) during the year 2022-23. The results embodied in the report have not been submitted to any other university or institute for theaward of any degree or diploma

INTERNAL GUIDE

Dr.R.V.S.S.S.Nagini

Asst.Professor

HEAD OF THE DEPARTMENT

Dr. Thayyaba Khatoon

CSE(AI&ML)

External Examiner

ACKNOWLEDGEMENT

We would like to express our gratitude to all those who extended their support and suggestions to come up with this application. Special Thanks to our mentor Asst. Prof. Dr. R.V.S.S.S. Nagini whose help and stimulating suggestions and encouragement helped us all time in the due course of project development.

We sincerely thank our HOD Dr. Thayyaba Khatoon for her constant support and motivation all the time. A special acknowledgement goes to a friend who enthused us from the back stage. Last but not the least our sincere appreciation goes to our family who has been tolerant understanding our moods, and extending timely support.

ABSTRACT

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 behavior of each malware on an emulated (sandbox) environment will be automatically analyzed and will generate behavior reports. These reports will be preprocessed into sparse vector models for further machine learning (classification). The classifiers used in this research are k-Nearest Neighbors (kNN), Naïve Bayes, J48 Decision Tree, Support Vector Machine (SVM), and Multilayer Perceptron Neural Network (MLP).The classifiers used in this research are k-Nearest Neighbors (kNN), Naïve Bayes, J48 Decision Tree, Support Vector Machine (SVM), and Multilayer Perceptron Neural Network (MLP). Based on the analysis of the tests and experimental results of all the 5 classifiers, the overall best performance was achieved by J48 decision tree with a recall of 95.9%, a false positive rate of 2.4%, a precision of 97.3%, and an accuracy of 96.8%. In summary, it can be concluded that a proof of-concept based on automatic behavior-based malware analysis and the use of machine learning techniques could detect malware quite effectively and efficiently.

1. INTRODUCTION

1.1 Problem Definition:

- Among the varied machine learning algorithms that are used for malware detection Naive Bayes, SVM and k-nearest neighbours have shown promising leads in malware detection
- We are using behaviour to detect the malware with the help of machine learning techniques.
- Now a days most of the antivirus softwares are using this behavior based malware detection.

1.2 Objective Of Project:

- Analysis of machine learning techniques used in behaviour based malware detection.
- Our main aim is to find malware in the system using behaviour of the system with the help of behaviour based model
- We can also detect unknown malware using behavior based model
- Scans the system in background automatically.

2. ANALYSIS

2.1 Introduction:

The problem to be examined involves the high spreading rate of computer malware (viruses, worms, Trojan horses, rootkits, botnets, backdoors, and other malicioussoftware) and conventional signature matching-based antivirus systems fail to detect polymorphic and new, previously unseen malicious executables. Malware are spreading all over the world through the Internet and are increasing day by day, thus becoming a serious threat. The manual heuristic inspection of static malware analysis is no longer considered effective and efficient compared against the high spreading rate of malware. Nevertheless, researches are trying to develop various alternativeapproaches in combating and detecting malware. One proposed approach (solution) is by using automatic dynamic (behavior) malware analysis combined with data mining tasks, such as, machine learning (classification) techniques to achieve effectiveness and efficiency in detecting malware 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 behavior of each malware on an emulated (sandbox) environment will be automatically analyzed and will generate behavior reports. These reports will be preprocessed into sparse vector models for further machine learning (classification)

2.2 Existing System:

Signature based malware detection .Antivirus products use signature-based detection in conjunction with a database. When scanning a computer, they’ll search for footprints matching those of known malware. If they discover one of these footprints, they’ll recognize it as malware, in which case they’ll either delete or quarantine it. User manually scans the files in the system using this model based software.

2.3 Proposed System:

- Behavior-based malware detection focuses on detecting intrusions by monitoring the activity of systems and classifying it as normal or anomalous. The classification is often based on machine learning algorithms that use heuristics or rules to detect misuse
- System automatically scans the systems without user permissions and throws notification to the user whenever a malware is found or footprint of malware is found.
- Only scans based on the behaviour of the system. When performance of the system reduced they this scans the files at back end and throws the notification to the user.

2.4 Software Requirement Specification:

2.4.1 Software Requirements:

During this project, we have used some software tools to perform the tasks of the modules.

2.4.1.1 Jupyter Notebook

2.4.1.2 PyCharm

2.4.1.3 Spyder

2.4.2 Hardware Requirements:

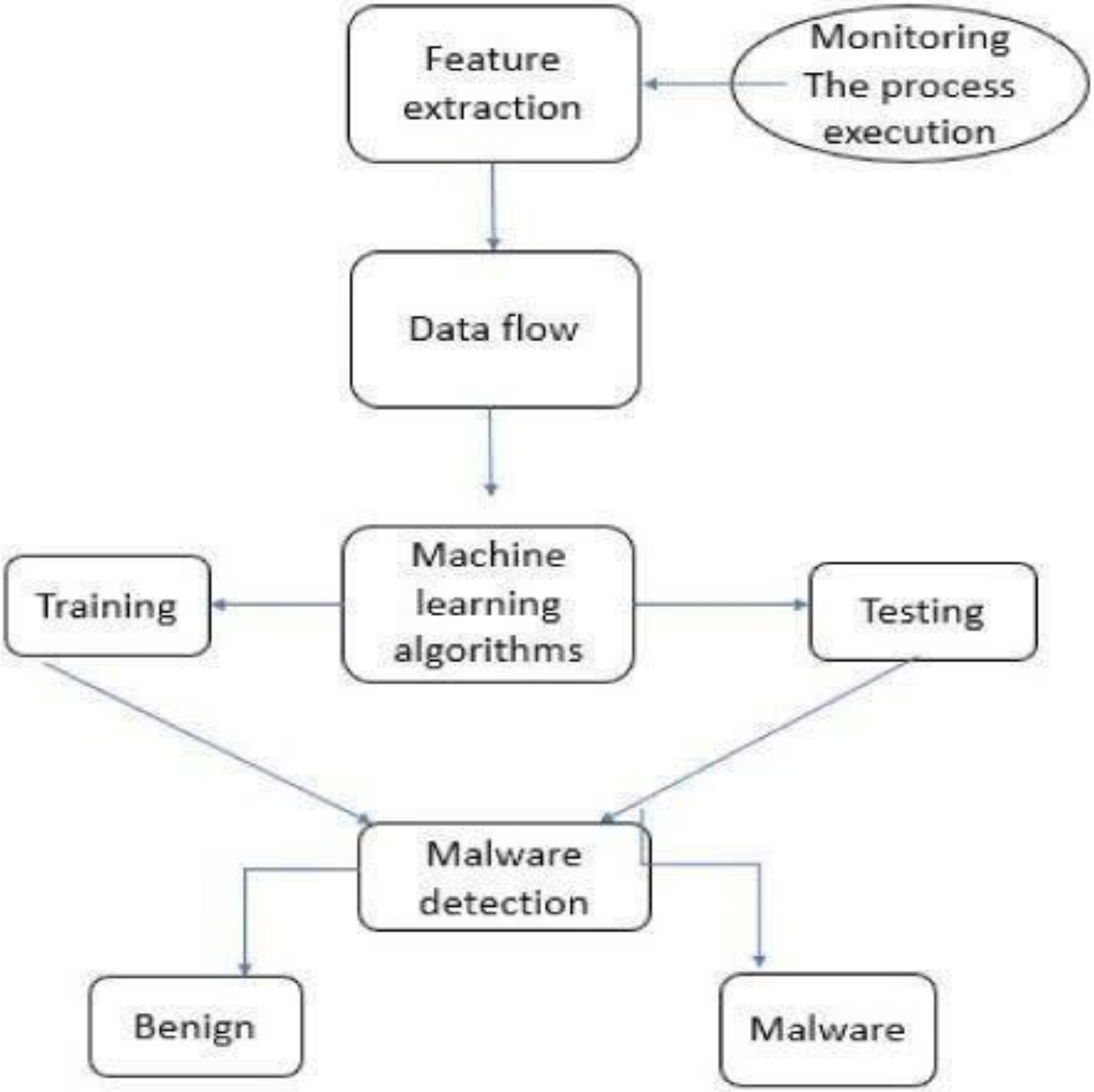
We performed this project based on some hardware requirements to better performance of the project.

- RAM : 4 GB
- ROM : 20 GB
- PROCESSOR : i5

2.5 Modules:

- Input the existing data set which is already categories as per the detected malware.
- Support vector machine(SVM)
- Naïve based algorithm
- K-Nearest neighbour algorithm

2.6 ARCHITECTURE:



DESIGN

3.1 INTRODUCTION

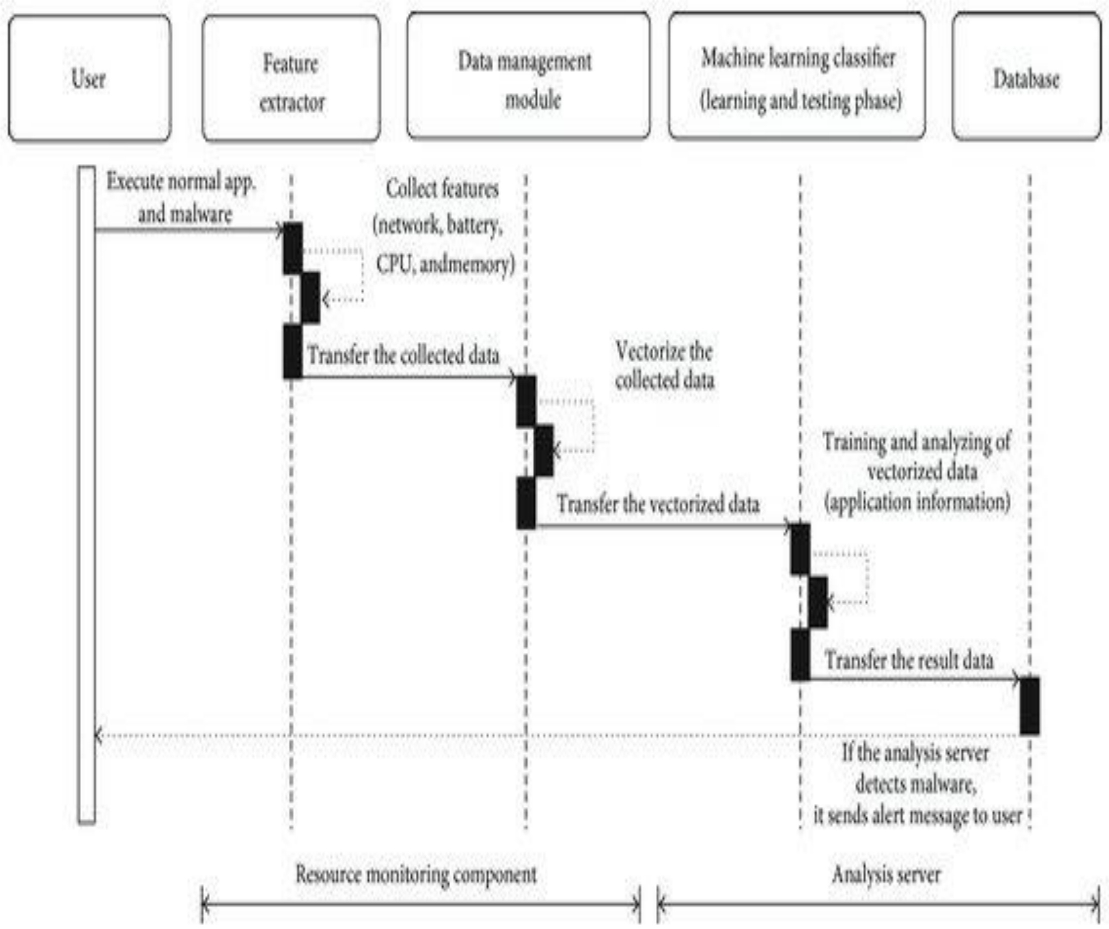
MALWARE is defined as software designed to infiltrateor damage a computer system without the owner’s in- formed consent. Malware is actually a generic definition for all kind of computer threats. A simple classification of malware consists of file infectors and stand-alone malware. Another way of classifying malware is based on their particular action: worms, backdoors, trojans, rootkits, spyware, adware etc. Malware detection through standard, signature-based methods is getting more and more difficult since all current malware applications tend to have multiple polymorphic layers to avoid detection or to use side mechanisms to automatically update themselves to a newer version at short periods of time in order to avoid detection by any antivirus software. For an example of dynamical file analysis for malware detection, via emulation in a virtual environment, the interested reader can see. Classical methods for the detection of metamorphic viruses are described.

An overview on different machine learning methods that were proposed for malware detection is given.

Those methods are:

- Naïve bayes theorem
- Support vector machine
- K-nearest neighbour

3.2 UML DIAGRAM OF MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES

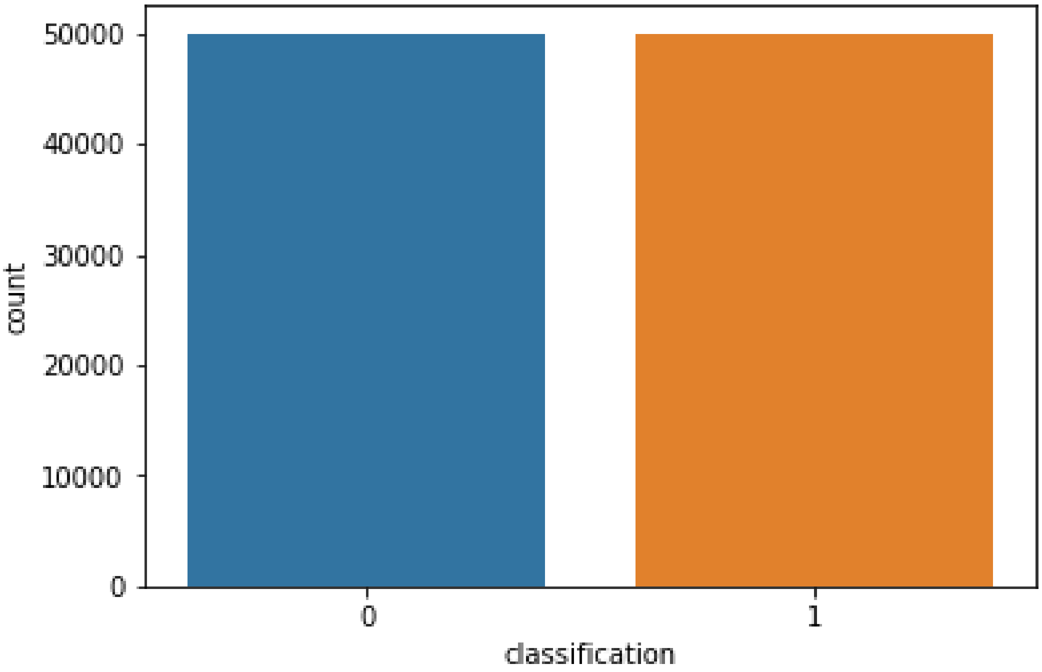

User
Feature extractor
Data management module
Machine learning classifier (learning and testing phase)
Database
Execute normal app. and malware
Collect features (network, battery, CPU, andmemory)
Transfer the collected data
Vectorize the collected data
Transfer the vectorized data
Training and analyzing of vectorized data (application information)
Transfer the result data
If the analysis server detects malware, it sends alert message to user
Resource monitoring component
Analysis server

3.3 DATASET DESCRIPTION

We have used a dataset containing both malware and benign values from different websites. We have selected some part of data for testing and some part of data for training. We divided the data using train_test_split method from sklearn module.

The number of malware files and respectively bening files in these datasets is classified under classification attribute/column. We classified the malware files and benign files and assigned 1 ,0 respectively and plotted a bar graph as shown in below figure.

We have removed unwanted columns of data from the data set for obtaining accuracy of the solution.



3.4 DATA PRE-PROCESSING TECHNIQUES

For obtaining a accurate value the data must be accurate and if the dataset containing any missing values or null values we perform pre-processing techniques. There are many techniques are available but we used only few techniques. They are:

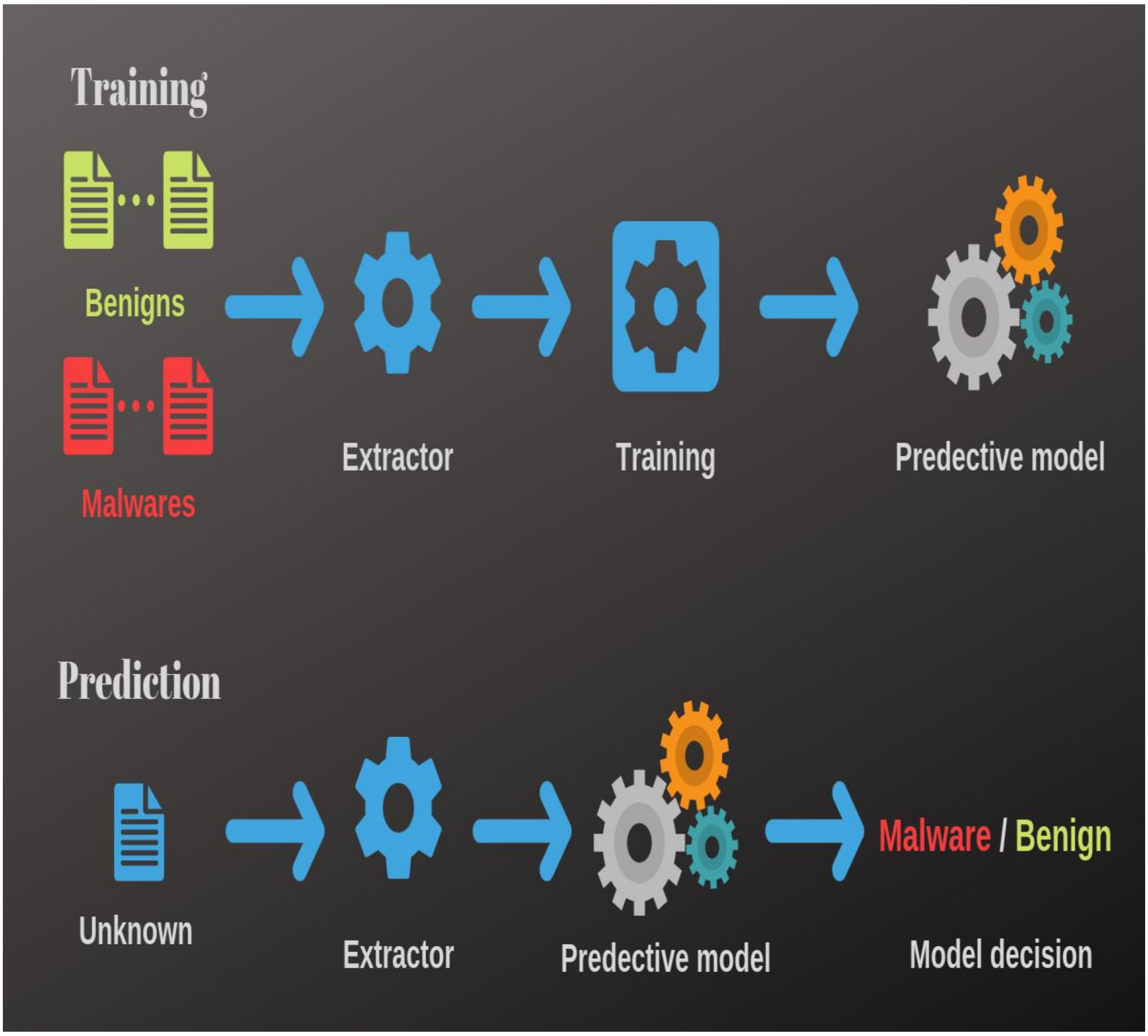
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set

3.5 METHODS AND ALGORITHMS

we are using three machine learning methods to detect malware using behaviour based model. The three methods are:

- Naïve bayes theorem
- Support vector machine
- K-nearest neighbour

3.6 BUILDING A MODEL



3.7 Evaluation

The evaluated results are:

- Data set is executed using naïve based algorithm and malware files and benign files are separated and a graph is formed between actuals values and predicted values of the data set

4.DEPLOYMENT AND RESULTS

4.1 INTRODUCTION

The source code is deployed into the Jupyter Notebook for the execution using the Decision tree and naïve bayes algorithm for malware detection using behaviour model.It is an open source that allows us to compile all aspects of a data project in one place making it easier to show the entire process of a project.It is very useful for creating and sharing documents

- It can illustrate the analysis process step by step by arranging the code.
- It is specifically designed for machine learning
- It quickly produce the output.It makes the output more efficient.

DECISION TREE

Decision tree 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

Decision tree is used to find the accuracy of the dataset. By using data pre-processing techniques it removes all the un wanted data and finds the accuracy of the dataset.

NAÏVE BAYES

Naïve bayes 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

Naïve bayes is used to find the accuracy of the dataset. By using data pre-processing techniques it removes all the un wanted data and finds the accuracy of the dataset.

4.2 SOURCE CODE

#NAIVE BAYES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data=pd.read_csv("MD.csv")
data.head()
data.shape
data.isnull().sum()
data.columns
data1=data.dropna(how="any",axis=0)
data1.head()
data1["classification"].value_counts()
data1['classification'] = data1.classification.map({'benign':0, 'malware':1})
data1.head()
data1.tail()
data1["classification"].value_counts().plot(kind="pie",autopct="%1.1f%%")
plt.axis("equal")
plt.show()
benign1=data.loc[data['classification']=='benign']
benign1["classification"].head()
malware1=data.loc[data['classification']=='malware']
malware1["classification"].head()
corr=data1.corr()
corr.nlargest(35,'classification')["classification"]
x=data1.drop(["hash","classification",'vm_truncate_count','shared_vm','exec_vm','nvcsw','maj_flt','utime'],axis=1)
x.head()
y=data1["classification"]
y
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()
model.fit(x_train,y_train)
pred=model.predict(x_test)
pred
model.score(x_test,y_test)
result=pd.DataFrame({
    "Actual_Value":y_test,
    "Predict_Value":pred
})
result
```

#DECISION TREE

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
pima=pd.read_csv("MD.csv")
pima.head()
pima['classification'] = pima.classification.map({'benign':0, 'malware':1})
pima.head()
col_names=['millisecond','classification', 'state', 'usage_counter','prio','static_prio']
pima.head()
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
clf=DecisionTreeClassifier()
clf=clf.fit(x_train,y_train)
y_pred=clf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test,y_pred))
```

4.3 FINAL RESULTS

Final results for the above code as follows:

Obtained accuracy through:

- Decision tree algorithm is 0.5932666666666667
- Naïve bayes algorithm is 0.6274

5.CONCLUSION

5.1 PROJECT CONCLUSION

Data set is executed using naïve based algorithm and Decision tree algorithm and malware files and benign files are separated and a graph is formed between actuals values and predicted values of the data set for naïve bayes and accuracy is obtained using Decision tree algorithm and the both accuracy is compared and there is a slight difference in the accuracy but both accuracy is almost equal.

REFERENCES

1.Charles lim,"Analysis of machine learning techniques used in behaviour based malware detection",Master of information technology department, Swiss German University

- https://www.researchgate.net/publication/232627329_Analysis_of_Machine_learning_Techniques_Used_in_Behavior-Based_Malware_Detection

2.N.Saravan "Malware detection using behaviour based model", member of Kaggle community.

- Malware Detection Using Naive Bayes | Kaggle

5.2 FUTURE SCOPE

Our future scope is to find more accuracy and develop an application which notifies the user about the malware in the websites. Many applications were already in the market but we want to find more accuracy in finding the malware and notifies the user.