International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 08 | August - 2022 Impact Factor: 7.185 ISSN: 2582-3930

MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES

A.GIRIJA[1],M.SABARI RAMACHANDRAN [2],N.BALASUBRAMANIAN[3],K.RAMYA[4]

1, student, Department of Master of Computer Application. Mohamed Sathak Engineering College. Ramanathapuram, India.

2, Assistant Professor, Department of Master of Computer Application, Mohamed Sathak Engineering College. Ramanathapuram, India.

3,Associate Professor, Department of Master of Computer Application, Mohamed Sathak Engineering College. Ramanathapuram, India

4,Assistant Professor, Department of Master of Computer Application, Mohamed Sathak Engineering College. Ramanathapuram, India.

ABSTRACT

Malware is malicious software disseminated to infiltrate the secrecy, integrity, and functionality of a system, such as viruses, worms, Trojans, backdoors, and spyware. To defend against an increasing number of sophisticated malware attacks, deep-learning based Malware Detection Systems (MDSs) have become a vital component of my economic and national security. The dataset, malware dataset is implemented as input. The input dataset is taken from dataset repository. Based on the characteristics of the observations, the dataset was created in a UNIX / Lunix-based virtual machine for classification purposes, which are harmless with malware software for Android devices. The data set consists of 100,000 observation data and 35 features. precision, recall, f1-score.

INTRODUCTION

Purpose

- Malicious files are being disseminated at a rate of thousands per day, making it difficult for this signature-based method to be effective.
- In order to combat the malware attacks, intelligent malware detection techniques need to be investigated.
- As a result, many researches have been conducted on intelligent malware detection by applying data mining and machine learning techniques in recent years.

PROJECT DESCRIPTION

- To classify or to detect the malware in the software.
- To implement the machine learning algorithms such as random forest and logistic regression.
- To classify or detect the malware effectively.

Input Data:

- The input data was collected from dataset repository.
- In this project, I have to use the malware detection dataset
- The dataset which contains the information about the classification(malware and benign) ,host etc.,
- My dataset, is in the form of '.csv' file extension.

Preprocessing:

- Data pre-processing is the process of removing the unwanted data from the dataset.
- Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0.

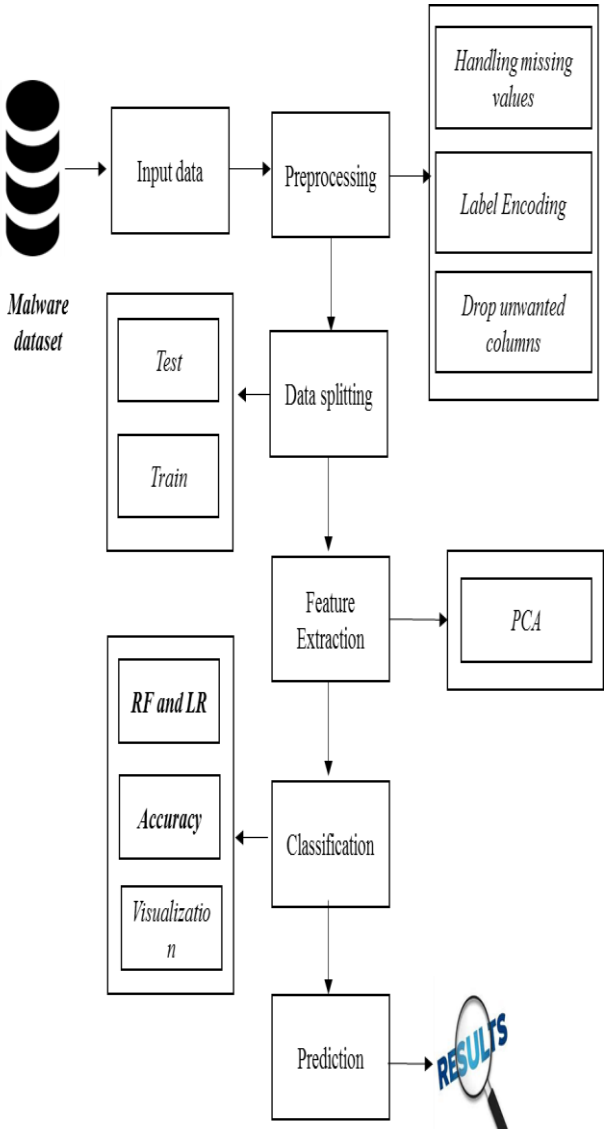
Data Splitting:

- In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how Ill it works.
- In my process, I considered 70% of the dataset to be the training data and the remaining 30% to be the testing data.

Classification:

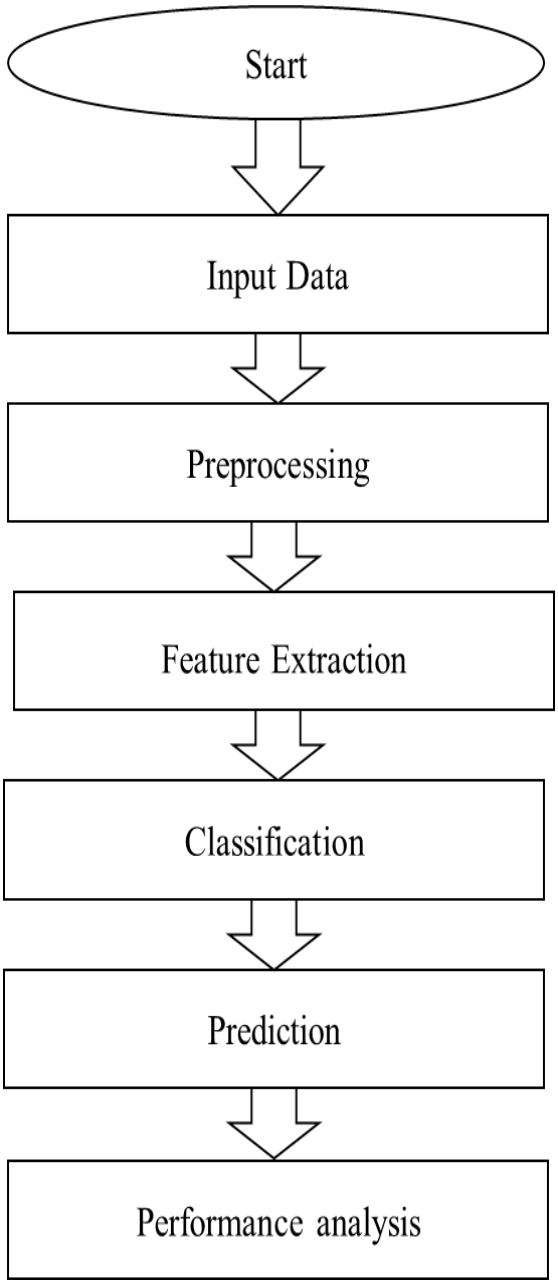
- In my process, I have to implement the two machine learning algorithms such as random forest and logistic regression.
- The **random forest** is a classification algorithm consisting of many decisions trees.
- **Logistic regression** is **a** statistical model that in its basic form uses a logistic function to model a binary dependent variable.

Flow diagram



System Architecture

/



SYSTEM IMPLEMENTATION

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error.

UNIT TESTING:

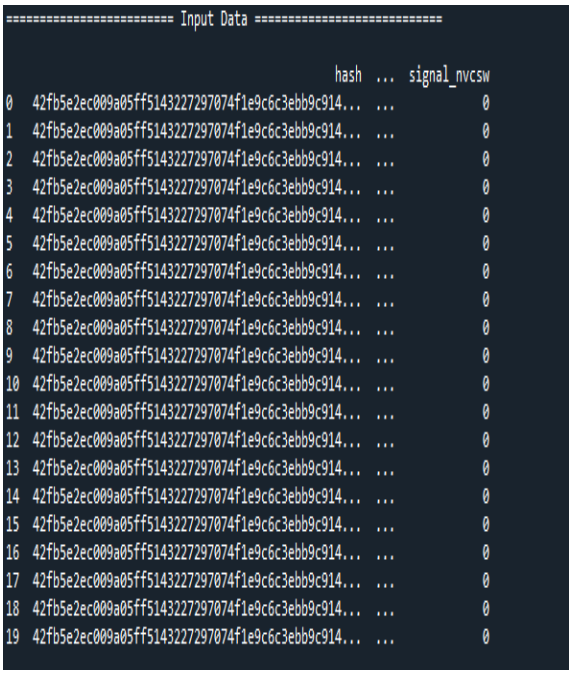
Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module.

WHITE BOX TESTING:

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases.

Screenshots:

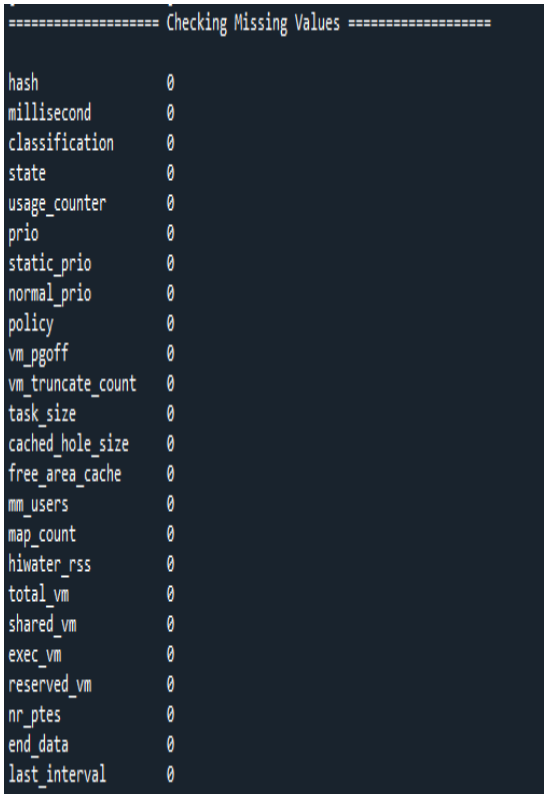
DataSelection:



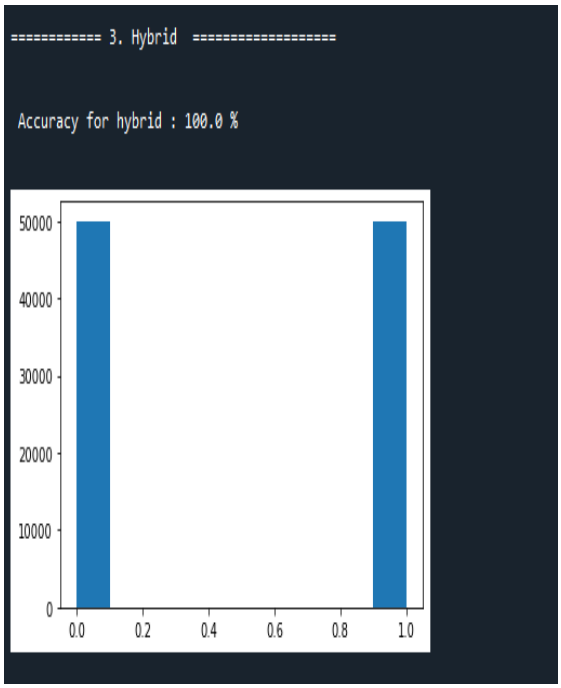
```
======================== Input Data ========================

                                                hash  ...  signal_nvcsw
0   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
1   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
2   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
3   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
4   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
5   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
6   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
7   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
8   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
9   42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
10  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
11  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
12  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
13  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
14  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
15  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
16  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
17  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
18  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
19  42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...  ...             0
```

Preprocessing:



COMPARISON:



CONCLUSION:

Finally, it provides a machine-learning based method for the detection of malware attacks in the software. Then, it provides good performance for both machine learning algorithms such as random forest and logistic regression.

FUTURE ENHANCEMENT

Further, new features could be added to the existing data, and these explorations are devoted for future research.

Future research entails exploration of these variations with new features that could be added to the existing data.

Future work since the malware defection is an important application in safety-critical environment.

REFERENCES:

M. Alazab, S. Venkatraman, P. Watters, M. Alazab, and A. Alazab, "Cybercrime: The case of obfuscated malware," in Global Security, Safety and Sustainability & e-Democracy (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 99, C. K. Georgiadis, H. Jahankhani, E. Pimenidis, R. Bashroush, and A. Al-

Nemrat, Eds. Berlin, Germany: Springer, 2012.

M. Alazab, "Profiling and classifying the behavior of malicious codes," J. Syst. Softw., vol. 100, pp. 91–102, Feb. 2015.

S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, and J. Yearwood, "Hybrids of support vector machine wrapper and filter based framework for malware detection," .

E. Raff, J. Sylvester, and C. Nicholas, "Learning the PE header, malware detection with minimal domain knowledge," in Proc. 10th ACM Workshop Artif. Intell. Secur. New York, NY, USA: ACM, Nov. 2017, pp. 121–132.

C. Rossow, et al., "Prudent practices for designing malware experiments: Status quo and outlook," in Proc. IEEE Symp. Secur. Privacy (SP), Mar. 2012, pp. 65–79. [11] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas. (2017). "Malware detection by eating a whole exe."

L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in Proc. 8th Int. Symp. Vis. Cyber Secur. New York, NY, USA: ACM, Jul. 2011, p. 4

L. Nataraj and B. S. Manjunath. (2016). "SPAM: Signal processing to analyze malware."

D. Kirat, L. Nataraj, G. Vigna, and B. S. Manjunath "SigMal: A static signal processing based malware triage," in Proc. 29th Annu. Comput. Secur. Appl. Conf. New York, NY, USA: ACM, Dec. 2013, pp. 89–98.

X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Proc. 14th Int. Conf. Artif. Intell. Statist., Jun. 2011, pp. 315–323