DETECTION OF MALWARE APPLICATIONS USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

CB.EN.U4CSE16008 B.Abhilash

CB.EN.U4CSE16032 N.Vamsi

CB.EN.U4CSE16035 P.Shanmukha Surya Sriram

CB.EN.U4CSE16048 P.Sanath Kumar

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE 641 112

OCTOBER 2019

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE, 641112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled " **Detection of Malware Applications using Machine Learning** " submitted by B.Abhilash (CB.EN.U4CSE16008), N.Vamsi (CB.EN.U4CSE16032), P.Shanmukha Surya Sriram (CB.EN.U4CSE16035) and P.Sanath Kumar (CB.EN.U4CSE16048) in partial fulfillment of the requirements for the award of the Degree **Bachelor of Technology in Computer Science and Engineering** is a bonafide record of the work carried out under our guidance and supervision at Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore

PROJECT GUIDE	CHAIRPERSON
Ms. Jeevitha K.P	**Dr. (Col) P.N. Kumar**
Assistant Professor	Professor
Dept. of Computer Science and Engg.	Dept. of Computer Science and Engg.

This project report was evaluated by us on :. . . . . .

INTERNAL EXAMINER EXTERNAL EXAMINER

Acknowledgment

We express our gratitude to our beloved Satguru Sri Mata Amritanandamayi Devi for providing a bright academic climate at this university, which has made this entire task appreciable. This acknowledgement is intended to be a thanks giving measure to all those people involved directly or indirectly with our project.

We would like to thank our Vice Chancellor Dr. Venkat Rangan. P and Dr. Sasangan Ramanathan Dean Engineering of Amrita Vishwa Vidyapeetham for providing us the necessary infrastructure required for completion of the project.

We express our thanks to Dr.(Col.P.N.Kumar), Chairperson of Department of Computer Science Engineering, Dr.C.Shunmuga Velayutham and Dr. G. Jeyakumar, Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. We express our gratitude to our guide, Ms. Jeevitha K.P , for the guidance, support and supervision. We feel extremely grateful to Dr. R.Gowtham, Dr. Dhanya N M, Ms. Anupa Vijay and Ms. K.Nalinadevi

for their feedback and encouragement which helped us to complete the project. We also thank the staff of the Department of Computer Science Engineering for their support. We would like to extend our sincere thanks to our family and friends for helping and motivating us during the course of the project.

Abstract

The number of malicious applications and adwares are increasing constantly on par with the number of devices. A great number of commercial signature based tools are available on the market which prevent to an extent the penetration and distribution of malicious applications. Numerous researches have been conducted which claims that traditional signature based detection system work well up to certain level and malware authors use numerous techniques to evade these tools. So given this state of affairs, there is an increasing need for an alternative, really tough malware detection system to complement and rectify the signature based system. Recent substantial research focused on machine learning algorithms that analyze features from malicious application and use those features to classify and detect unknown malicious applications.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

abbreviation FullForm

Chapter 1

Introduction

1.1 Background

Malware is defined as software or piece of code which can infiltrate or damage a system without the owner's consent. It is becoming increasingly difficult to detect malware using old signature-based methods as most of the current malware are either polymorphic wherein each copy of virus uses a different key or are metamorphic wherein each new version uses non-cryptographic obfuscation and thereby making it more difficult to get signature. The basic idea of any machine learning task is to train the model based on some algorithm and perform classification or predict new values. Training is done on the input dataset and the model is built which is then subsequently used to make predictions. Malware detection is a classification problem. We can train a program to recognize whether a piece of software is a malware or not and thus we can then detect later that whether a given file is malware or not.

1.2 Problem Statement

Most malware detection methods are based on traditional content signature based approaches in which they use a list of malware signature definitions, and compare each application against the database of known malware signatures. The disadvantage of this detection method is that users are only protected from malware that are detected by most recently updated signatures, but not protected from new malware(i.e. zero-day attack). A previous study of the malicious patterns has concluded that "Signature-based approaches never keep up with the speed at which malware is created and evolved". In this thesis, our goal is to find a solution that can process an application, extract features and try to predict whether the application under process may be Malware or Benign using machine learning.

1.3 Specific Objectives

In this project, our main objective is to detect whether the given application is malicious or not by using machine learning techniques. Analyzing the performance of each machine learning algorithm for the same dataset which contains malicious applications.

1.4 Findings

Through our project, we have successfully tested whether the considered application is malicious or not and also analyzed how each machine learning algorithm fares for the same malicious application filled dataset.

Chapter 2

LiteratureSurvey

The following section provides a review of the literature related to the malware detection using machine learning techniques. The trained model developed using machine learning algorithm is used to detect whether the given application is malicious or not.

The development was largely based on the methods discussed in 'MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES' by Rameez Raza. The author provided a detailed research on the existing techniques of malware detection and the problems with the current techniques. The author also explained how machine learning is used to classify the malicious applications.

Chapter 3

Proposed System

3.1 System Architecture

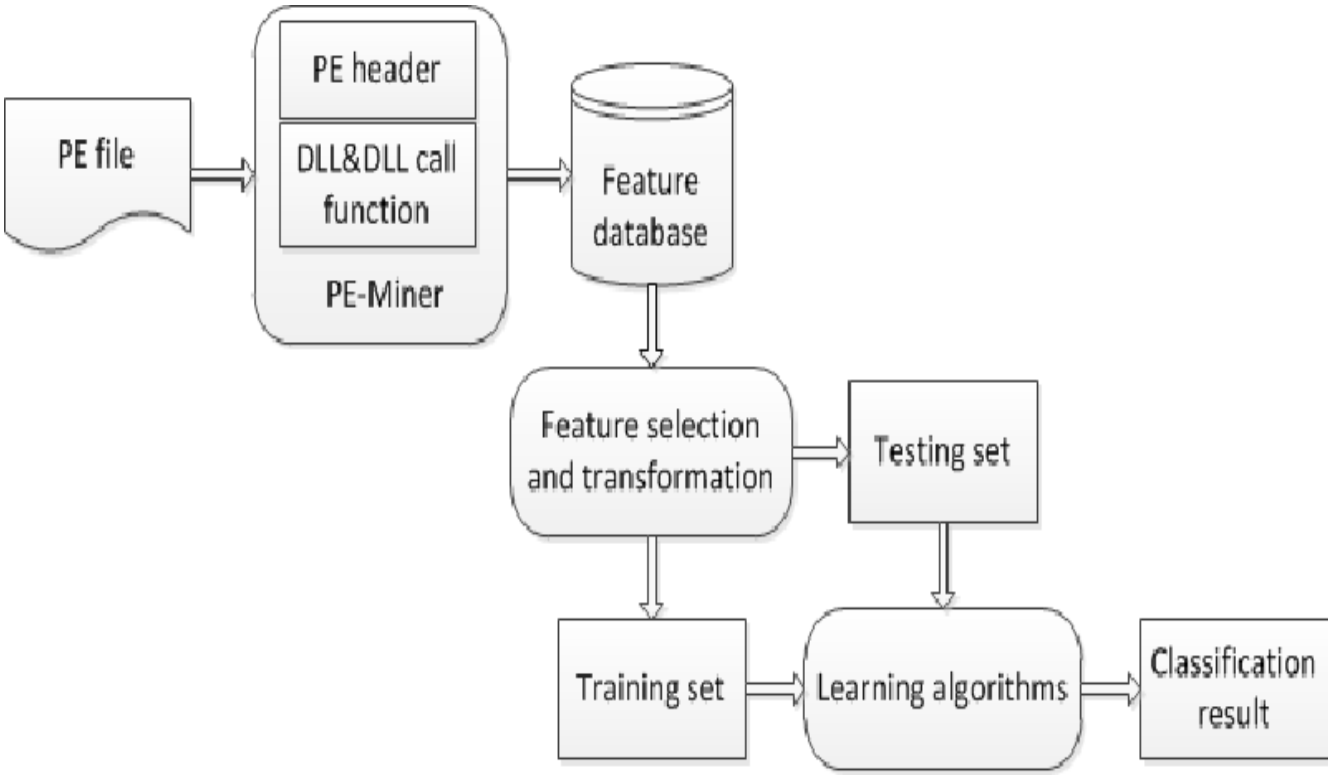

Figure 3.1: System Architecture Diagram

3.2 System Specification

3.3 Methodology

The key identifier of malware is that it either changes the control stream or do change to the information structure. These progressions impact memories get to design by the myogram. Our exploratory assessment centers around Portable executable (PE) document's header fields esteem. We removed all header information from PE informational index. We classified the heaviness of each component in order to guarantee that just imperative highlights that guide in our investigation are at last considered. The proposed learning display for malware location has four noteworthy targets as recorded beneath: 1. Feature set extraction from PE header. 2. Select relevant headers and modify/derive few features thereby enhancing the detection capability of malware. In the paper, they have considered only one classifier to find feature importance but we used two classifiers and combined their results to find relative importance. 3. Application of machine learning techniques on the complete feature set and develop a model for classification. 4. Apply test data on the model and verify the result. Additionally, we analyzed false positives and false negatives to understand file header values of malicious files and we also tried to modify PE header values of few malicious files to check our classifier accuracy.

1. The training dataset is taken and a set of features are extracted based on training scores.

2. The machine learning model is trained on this training dataset based on features selected in step 1.

3. The trained model is applied to test dataset and the accuracy of the model is calculated.

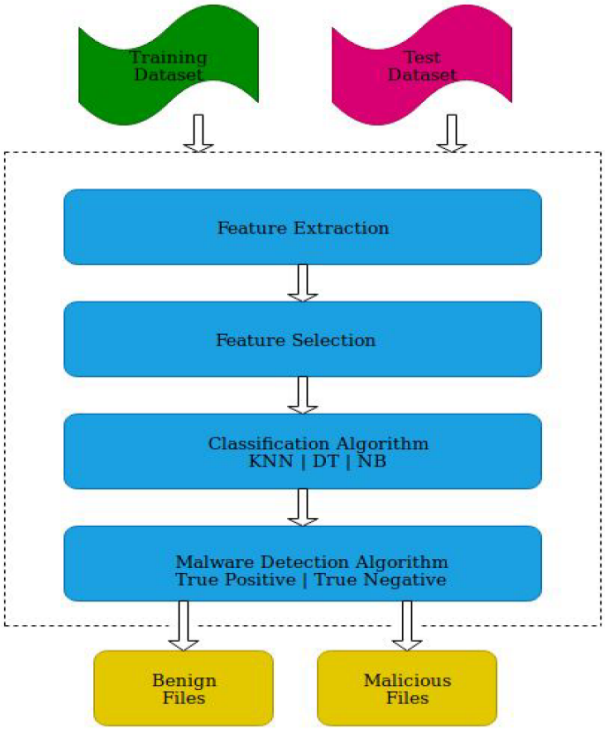

Figure 3.2: Flowchart of Methodology

3.4 Implementation

1. Getting dataset from http://ocslab.hksecurity.net/apimds-dataset

2. Use extra trees classfier and get the importance of features.

3. Use cross validation to split dataset into train and test data.

4. Use 6 methods/ algorithms to find accuracy Using 6 models below : Decision Tree Random Forest Adaboost Gradient Boosting GNB : GaussianNB Linear Regression

5. Saving models that is objects using 'joblib'.

6. Getting features from unseen file using pe library.

7. Testing using model formed.

Chapter 4

Results and Discussion

On running different machine learning algorithms on the taken malware dataset, the accuracy of each algorithm is displayed as follows :

```
results = {}
for algo in model:
    clf = model[algo]
    clf.fit(X_train,y_train)
    score = clf.score(X_test,y_test)
    print ("%s : %s " %(algo, score))
    results[algo] = score
```

```
RandomForest : 0.994386091996
GradientBoosting : 0.988373777617
GNB : 0.702897500905
DecisionTree : 0.990981528432
LinearRegression : 0.54036008649
Adaboost : 0.986381745744
```

Figure 4.1: Accuracy of machine learning algorithms

By loading python file in the code and running, it classifies whether the

taken file is malicious or not.

```
%run malware_test.py "/home/Downloads/Skype.exe"
```

```
The file Skype.exe is legitimate
```

Figure 4.2:

```
%run malware_test.py "/home/Downloads/BCN12ui49823.exe"
```

```
The file BCN12ui49823.exe is malicious
```

Figure 4.3: Classification of the file type

Chapter 5

Conclusion and Future Work

1. The proposed machine learning technique used a static analysis technique to extract the features which have low time and resource requirement than dynamic analysis. Better classification accuracy can be achieved by building malware classifier using header fields' value alone as the feature.

2. The real-world scenario is unpredictable and can be different than the experimental environment so to protect a sensitive system from malware, it is not advisable to use only headers values based classifier. For example, a carefully crafted malware would have a benign header and malicious payload hidden in the body of PE file.

3. We have experimented with PE file format.

Chapter 6

Bibliography

[1] Ajit Kumar, K S Kuppusamy, and Aghila Gnanasekaran. A learning model to detect maliciousness of portable executable using an integrated feature set. In the Journal of King Saud University - Computer and Information Sciences, 01 2017.

[2] D. Gavriluţ, M. Cimpoeşu, D. Anton, and L. Ciortuz. Malware detection using machine learning. In 2009 International Multiconference on Computer Science and Information Technology, pages 735–741, Oct 2009.

[3] Swapnaja Hiray Smita Ranveer. Comparative analysis of feature extraction methods of malware detection, June 2015.

[4] Mihai Christodorescu and Somesh Jha. Static analysis of executables to detect malicious patterns. In Proceedings of the 12th Conference on USENIX Security Symposium Volume 12, SSYM'03, pages

12–12, Berkeley, CA, USA, 2003. USENIX Association.

[5] S. Y. Yerima, S. Sezer, and I. Muttik, Android Malware Detection Us- ing Parallel Machine Learning Classi

ers, in 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies, 2014.

[6] Mas ud, Mohd Zaki, (2014), Analysis of features selection and machine learning classifier in android malware detection, Information Science and Applications (ICISA), 2014 International Conference on. IEEE.