A

PROJECT-REPORT

on

“Classification of Malwares using Machine Learning”

Submitted in partial fulfilment for the requirements of the Doctoral course work in Machine Learning (CS 403/603)

In the Discipline of

COMPUTER SCIENCE & ENGINEERING

Indian Institute of Technology Indore

By

Ms. Sharmila SP (2201101012)
Mr. Shekhar Tyagi (2201101013)

Under the guidance of

Dr. Puneet Gupta
Assistant Professor

Department of Computer Science & Engineering
Indian Institute of Technology, Indore, Madhya Pradesh

2022-23

Identification and Classification of Malwares

Sharmila S P and Shekhar Tyagi

10 November 2022

Abstract: Malware is one of the major obstruction for the growth and increase of economy of a country, as most of the people are continuosly connected over internet, we are under threat.

Malware is a small program which is intentionally written to indulge some malicious activities in to a targeted system. the intention of coding malware is to gain unauthorised access into a system. Malware exhibits dynamic behavior. Some of the features on which its behavior depends on are the technology with which it is designed, nature of the attack, and the exploitation of the vulnerability. Along with the rapid growth in the use of digital devices and the Internet, the number of malware attacks has been increased. Since from last five years several cyber criminals are engaged in creation of these malwares. Thus malware detection is receiving interest from researchers. Malware development made with sophisticated techniques that bypass even the most advanced digital security systems. In this context, we endorse a classification system to classify a given sample as normal or malware with good accuracy and low computation cost.

1 Introduction

1.1 Malware

Malicious application is an deliberately written application to bask in various benign activities, From person's information stealing to cyber-theft. The act exposed by way of the malware is depending on different factors inclusive of type of the attack, sophisticated generation and the speedy growth in reluctant vulnerabilities[1]. Malware assaults also multiplied to the rapid increase within the use of virtual devices and internet. The incremental boom within the creation of latest malware in the near 5 years made the detection as a attractive problem for researchers. Malware development is pretty organized with constant innovation, and sophisticated strategies are continuously being evolved to bypass even the most superior virtually based security systems[2]. Computationally unbounded adversary is constantly coding such malwares. So we need an efficient system for classifying a file as malware or not malware.

Different editions of malwares consist of the following:

- Trojans
- Back door
- Ransomware
- Spyware

Some of the strategies with which malwares spread and multiply:

- Repackaging
- Drive by way of download
- Dynamic payloads
- Stealth malware strategies

1.2 PE file

Due to the extremely good reputation of the Windows working system, Portable Executable (PE) documents had been at the centre of the focus of organized cybercrime companies for several years now, PEs are executable record codecs or object code including .exe, .dll, .sys, .ocx, and .drv, used in 32/64-bit versions of the Windows running machine. The common data structure used in windows loader for managing and running executable code contained in each file is *eir format[3]. PE archetype has a couple of headers and dynamic linkers. these are useful when a file is allocated memory in a system. this executable chain has some regions with some requirements for memory protection techniques.

Windows NT has introduced these PE files for the first time which are specific type of windows format files. This is the standard binary format for EXE and DLL files on Windows. Since its introduction extensions have been created for other binary formats including .NET and 64-bit support (PE+ 32)[4]. The PE file begins with an MS-DOS header, the bytes being an ASCII MZ value as shown in Figure 1.

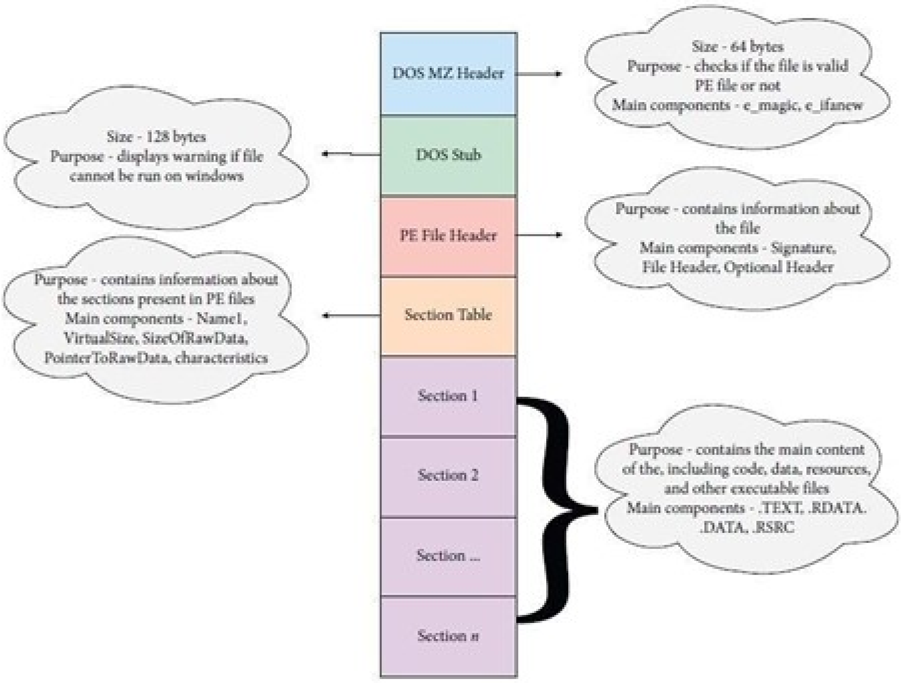
DOS MZ Header
Size - 64 bytes
Purpose - checks if the file is valid
PE file or not
Main components - e_magic, e_ifanew
Size - 128 bytes
Purpose - displays warning if file
cannot be run on windows
DOS Stub
PE File Header
Purpose - contains information about
the file
Main components - Signature,
File Header, Optional Header
Purpose - contains information about
the sections present in PE files
Main components - Name1,
VirtualSize, SizeOfRawData,
PointerToRawData, characteristics
Section Table
Section 1
Section 2
Section ...
Section n
Purpose - contains the main content
of the, including code, data, resources,
and other executable files
Main components - .TEXT, .RDATA.
.DATA, .RSRC

Figure 1: Basic structure of PE file

An advesary may insert any any snippet he intend to in the malware. the offset to the PE signature is available at the address 0x3c. The format of PE signature is same as MS-DOS stub. it has 4 bytes to identify image format. This image file has P,E and two bytes then followed by PE header. There is another header Coeff header which includes machine field. We will be working on intel 386 processoror compatible. the first part of the PE file coeff header is another optional header.

This header is divided into two parts: window-specific fields and data directories. Window-specific fields are used by the operating system to load and run the file. Data directories contain the location (RVA) and size of various tables required by the operating system, these are loaded into memory at runtime.

1.3 Motivation

Due to the increasing demands and popularity of web applications and consistent usage of internet, an extremely good reputation of the Windows working system, Portable Executable (PE) documents had been at the centre of the focus of computationally unbounded cybercriminals for several years now. Thus there is an urgent need to detect the malwares. Once such malware is injected into your device, may it be a mobile or stand alone PC, you are compromised with the attacker. All your information can be easily accessed by him without your knowledge. This motivated us to work on this problem. Here we make a small attempt to contribute to the digital world by devicing a classification model to classify the files as normal or malware.

1.4 Problem Statement

Malicious software or Malware is software this is programmed to harm or create issues for the user or the machine. This malicious software is surely an growing risk to computer systems and networks which might be owned by using large organizations and huge corporations. And due to those troubles malware evaluation and detection have emerge as a key problem in nowadays era. It is a reality that a plethora of malicious software program like laptop viruses has been created in latest years.

As we've visible the unique non-stop evolutions inside the discipline of cyber securities and despite the ones good sized enhancements in security mechanisms, malware continues to be amongst the most important chance to our on-line world. Facing this trouble many researchers and providers are coming collectively towards a way to discover a quicker alternate approach of detection and evaluation of malware that could grow to be a savage to protect the modern day internet world from being attacked.

1.5 Objective

The goal of our project is to read files from the dataset and classify it as legitimate file or a malware. Here we employ Random Forest technique for implementing the task. Malware detection refers to the process of detecting the presence of malware in a PE file or distinguishing whether a selected program is malicious or benign (valid).

1.6 Hardware and Software Specification

Hardware Requirement:
Processor : i5-10300H CPU 3.65GHz 2.50 GHz.
Hard drive : 500MB
Installed RAM : 32.00 GB
System type : 128-bit operating system, x128-based processor.

Software Requirement:
Windows 10 or 11 64-bit OS
Languages used: Python, JavaScript, HTML, CSS

2 Literature Review

The process of distinguishing deficious from benign applications. Both the methods of sophistication and anti-malware techniques have been collectively in an arms race of compressing every different detection strategies by way of its form of evaluation are categorised into statical and dynamical. Static evaluation does no longer execute the sample whereas dynamic analysis executes the sampling in a established surroundings. A good amount of works have best considered the bare fee of PEs area as capabilities and used them with other complementary functions like .dll and various APIs calls are extracted from recorded n-directories, or capabilities constructed using sections header. The paintings makes use of a fixed of capabilities together with entropy, packer data, remember of suspicious sections, and so forth. Which might be calculated on the premise of header subject cost.

PCI Android
Author: Vikas Sihag , Gaurav Choudhary , Manu Vardhan.
Published in: 2019.
Method: Monkey tool.
Results: Performed much better than the other techniques and the Monkey achieved the best segment coverage.
Issues: Proposed approach suffers from few limitations-coverage issues, random events in emulator.

Review on malware detection techniques
Author: VMER ASLAN AND REFIK SAMET (IEEE Members)
Published in: 2019
Method: Data mining and ML algorithms.
Results: Detecting traditional and new generation malware.
Issues: Detection for varied apps still remain a challenging Task.

Integrated static and dynamic analysis for malware detection
Author: P V Shinjo, A.S. Salim
Published on: 2020.
Method: API calls.
Results: 98.4% accuracy utilising in the best case via random forest (RF) algorithm.
Issues: Less accuracy in static methods.

3 PROPOSED SYSTEM

3.1 Data Flow Diagram

In order to achieve the target objective the samples are accumulated through Dataset acquired via Kaggle website. Raw samples were processed before the learning and feature extraction. The duties in the pre-processing step are document entity identification, duplication elimination and labellisation. Labelling is an essential project. We labelled every pattern in both malware and benign. Labeled samples are then handed to characteristic extracting step wherein PEs header subject's data is used to evaluate raw and included set of features. With these characteristic set, exclusive learning based algorithms are skilled were examined in closing step, Feature based generations is an essential task.
This task is finished in four essential processes:
(1) Raw data collection.
(2) Pre-data processing.
(3) Feature extraction.
(4) Training
(5) Classification using Random Forest

Samples were gathered from various resources and are saved one at a time in additional processes. Raw data samples must be processed to make usage of gadget gaining knowledge of education.

First step in which PEs head discipline's values are extracted to create uncooked and integrating feature set. With uncooked and integrated feature based set, unique device learning based algorithms are skilled and examined in the last step All labeled malware and benignated samples along side sub header rules are enter to the set of rules. Two loops, one for benign and the other for malware samples run to carry pattern after the other for processing.

4 IMPLEMENTATION

The malware data set was downloaded from Kaggle.com this work compared and applies varied machine learning based algorithms to classify.
The following methods were employed in the literatures we referred.
o Naive Bayes
o Decision Tree
o Ada Boost
o Gradient Boost
But we are using Random forest in our implementation work:

Random Forest had the best AUC result of 0.994 for varied -class and 0.998 for bin classifications. Extra Trees Classifier facilitates a number of randomized choice trees with diverse meta-samples of the dataset and use submissive averages to improve the predictive accuracy and manipulate overfitting. Extra Trees Classifier helps in choosing the desired functions beneficial for classifying a document as both Malicious or Legitimate.

14 features are identified as required by extra Trees Classifier
1. Dll Characteristics
2. Characteristics
3. Machine
4. Sections Max Entropy
5. Major Subsystem Version
6. Resources Min Entropy
7. Resources Max Entropy

8. Image Base
9. Version Information Size
10. Size Of Optional Header
11. Sections Mean Entropy
12. Subsystem
13. Main Operating System Version
14. corrective tag

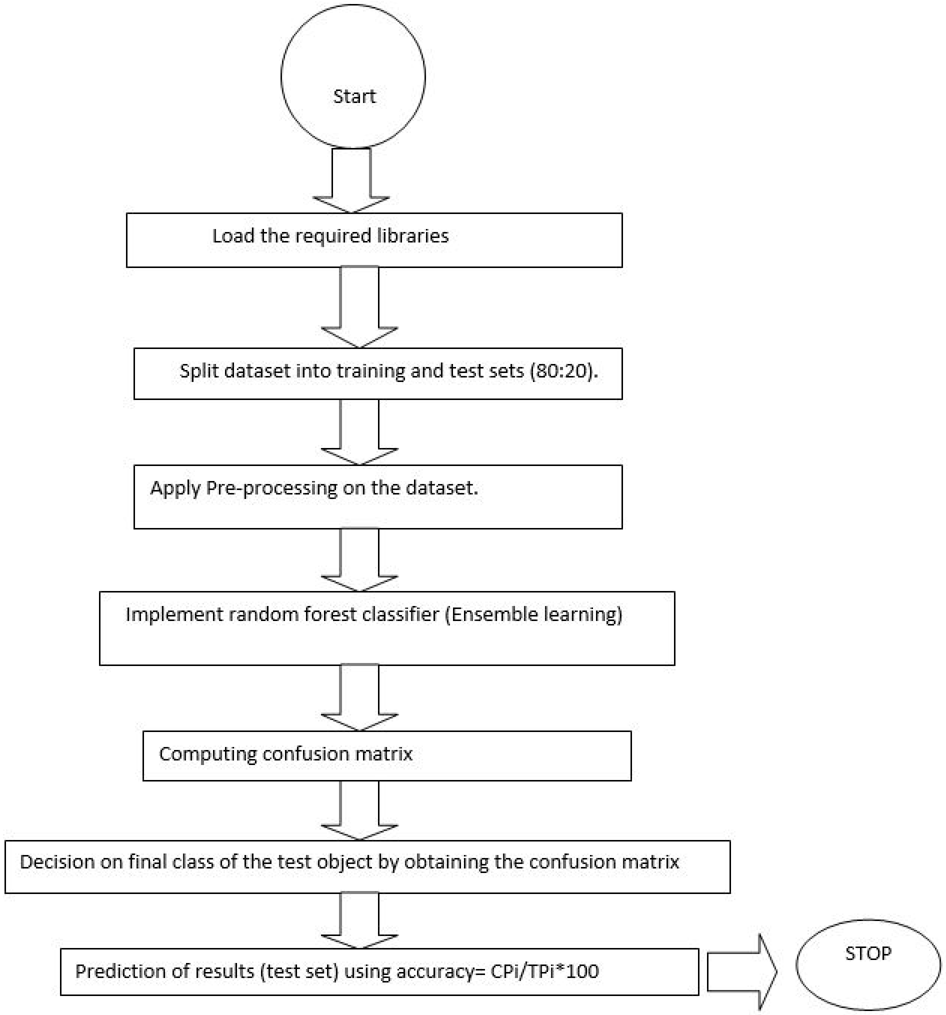
Start
Load the required libraries
Split dataset into training and test sets (80:20).
Apply Pre-processing on the dataset.
Implement random forest classifier (Ensemble learning)
Computing confusion matrix
Decision on final class of the test object by obtaining the confusion matrix
Prediction of results (test set) using accuracy= CPi/TPi*100
STOP

Figure 2: Dataflow Diagram

Random Forest:
Using the available dataset several decision trees are constructed using Random forest technique. Random set of elements are selected whenever there is a split at each node. we use the average probability from each decision tree to perform this classification

Here is the basic algorithm for training a Random Forest classifier For each decision tree that will be grown: Choose a random set from the given data set, this can be used to create every selection tree.
Expand every tree, observe given steps on each node until the minimum depth is achieved.
i. Randomized pick out a hard and fast of feature sets.
ii. Select the satisfactory cut up primarily based on the randomized one decided on capabilities.
iii. Split the node into binary infant nodes.

5 Result analysis

Training-test break up is a way of splitting the dataset. It takes the percentage as dividing threshold. Training dataset is used for training the system understand the algorithm whilst checking out dataset tests the performance.

We use 80: 20 ratios for splitting our uncooked and incorporated function and for this reason 20% of the dataset used to educated category algorithms and 20% is used for checking out.

Machine learning methods are highly beneficial when it is required to increase the efficiency of classification of malwares. Furthermore the performance of this model can be increased by doing Hyper parameter Tuning using GridsearchCV. As its hard for ML algorithm to work with categorical data so it is required to convert into numerical data hence categorical parameters in case of 1 hot encoding will prepare separate columns for X and the label encoding is used for y. The accuracy for the proposed model implemented using random forest comes to be 99.51%

6 Conclusion

Detecting a malicious PEs file from benign may be very crucial as PEs format is the well understand norm format, utilized in Win- OS. our design presents a singular derivable characteristic with engineered method that increase the performance of device gaining knowledge of a ML based classifier for malicious PE record detection. Here we practice static evaluation methods to extract the features which consumes much less time and aid requirement than viable analysis. This work also provides important evidence for adaptability of various headers fields cost of PEs file as useful disccriptive functions. Achieved results virtually propose that values obtained from these may be inculcated to classify record is malware or benign.

REFERENCES

[1] Schacht and P. Kieseberg, "An analysis of 5 million openpgp keys," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA),vol. 11, no. 3, pp. 107–140, 2020.

[2]S. Wang, Q. Yan, Z. Chen, B. Yang, C. Zhao, and M. Conti,"Detecting android malware leveraging text semantics of network flows," IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1096–1109, 2017.

[3] Zulkifli, I. R. A. Hamid, W. M. Shah, and Z. Abdullah,"Android malware detection based on network traffic using decision tree algorithm," Advances in Intelligent Systems and Computing, Springer, in Proceedings of the International Conference on Soft Computing and Data Mining, pp. 485–494,January 2018.

[4]A. F. A. Kadir, N. Stakhanova, and A. A. Ghorbani, "An empirical analysis of android banking malware," Protecting Mobile Networks and Devices: Challenges and Solutions,Vol. 209, CRC Press, , Boca Raton, Florida, 2016.

************ End of the Report ************