Detection of Malware Using Machine Learning Algorithms

Taha HARBOUCH

Higher School of Technologies Essaouira (ESTE) Cadi Ayaad University, Marrakech, Morocco

Abstract: This pepper discusses the usage of machine learning in malware detection, dataset used contained raw data of more than 21600 file to train and test four machine algorithms: k-nearest neighbors, Random Forest, Support Vector Machine and , Naïve Bayes. The performance of each algorithm in classifying malwares is based on the accuracy, detection rate and false alarm rate.

Keywords: Malware detection, Machine learning, SVM, Random Forest, Naïve Bayes, KNN

1. Introduction

Beside the growth of the technology, still the security of the computer makes people worried. The security of the computer system becomes more vulnerable than ever the technological progress brings the complexity of the application environment, protecting computer on a network is hard due the open characteristic of the internet therefore malware has stronger destructive ability. Malware "is defined as software designed to infiltrate or damage a computer system without the owner's informed consent. Malware is actually a generic definition for all kind of computer threats." [1]. The Current antivirus software are unable to detect zero-days attacks - malicious attacks targeting vulnerabilities that are previously undisclosed- because they rely on the principle of signature-based methods - signature is unique hex code strings in each malware or infected files- [2], in order to create a more robust and reliable antivirus product, we need to develop alternatives that complement traditional signature-based detection. To solve this problem, machine learning methods and classifiers are used for computer malware detection [3].

In recent years various research have focused on developing classification and clustering techniques to automatically categorize malware into malware families here are some researchers for malware classification and detection in [4].

In this work a machine learning approach was used for malware detection. For this purpose, 140849 legitimate samples and 75503 malwares simples this data was obtained from the malware security partner of Meraz'18 - Annual Techno Cultural festival of IIT Bhilai, the said raw data constituted malware and legitimate files, Statistical analysis was done on these files which mainly constituted the extraction of PE information and calculation of entropy of different sections of these files [5].

The rest of paper is organized as follows. In Section 2 briefly discuss PE format and the differences between benign software and malware. Section 3 presents the architecture of the approach. Section 4 and 5 discuss and compare algorithms results. Finally, we present our conclusions in Section 7.

2. PE File Format

The Portable Executable (PE) format is a file format for executables, object code, DLLs, FON Font files and others used in 32-bit and 64- bit versions of the Windows operating system. The PE32 format stands for Portable Executables of 32-bit while PE32 + stands for Portable Executables of 64-bit format.

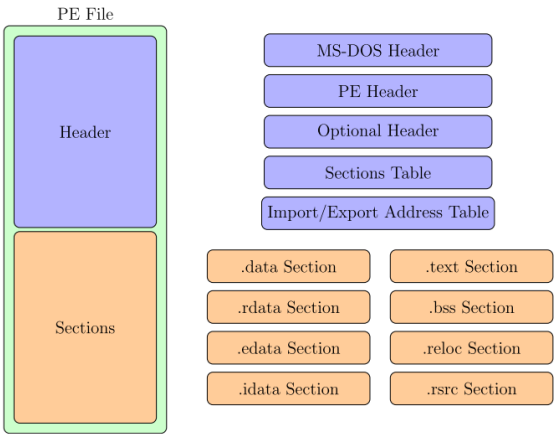

Figure 1: PE file architecture

The format information of PE file is illustrated in Figure 1. It is basically a data structure that encapsulates the information necessary for the Windows OS loader to manage the wrapped executable code. A PE file consists of a PE file header and a section table (section headers) followed by the sections' data. The PE file header consists of a MS DOS header, the PE signature, the image file header, and an optional header. The file headers are followed immediately by section headers. Section header provides information about its associated section, including location, length, and characteristics. Section is the basic unit of code or data within a PE or COFF file. Different functional areas, such as code and data areas, are separated logically into sections. In addition, an image file can contain a number of sections, such as.tls and.reloc, which have special purposes.

Many fields of PE file have no mandatory constraint. There are a number of redundant fields and spaces in PE file, so that it has created opportunities for malware's propagation and hide. It also makes the format information of malware and benign software show many differences. Although format information is not very advanced, it is an effective way to detect even polymorphic malware. In summary, there are many differences in format information between malware and benign software. The application of data mining methods to study these differences of format information is a feasible way to detection of known and unknown malware. More information on the PE file format can be found in the documentation provided by Microsoft.[6]

3. Architecture of the approach

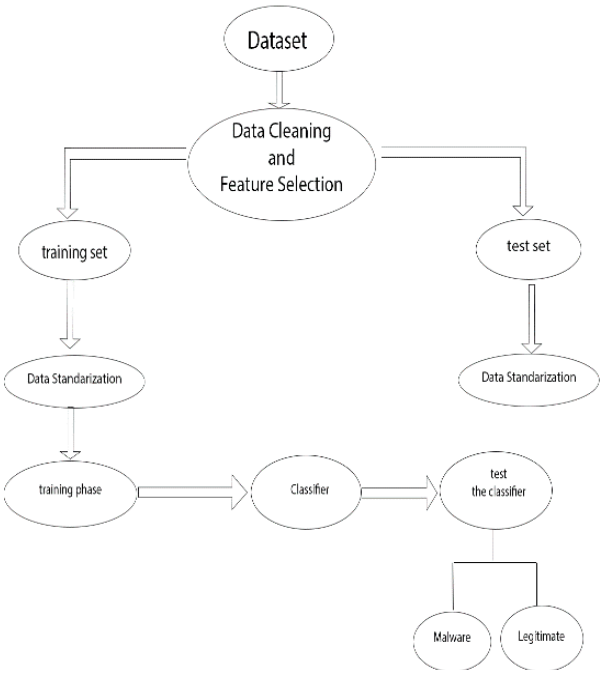

Figure 2: Architecture of the approach

a-Dataset Features:

As mentioned in the introduction the used dataset contained raw data of PE information which was extracted from malware and legitimate files. And here down below the list of extracted features:

md5 Machine, SizeOfOptionalHeader, Characteristics, MinorLinkerVersion, SizeOfCode, SizeOfInitializedData, SizeOfUninitializedData, AddressOfEntryPoint, BaseOfCode, BaseOfData, ImageBase, SectionAlignment, FileAlignment, MajorOperatingSystemVersion, MinorOperatingSystemVersion, MajorImageVersion, MinorImageVersion, MajorSubsystemVersion, MinorSubsystemVersion, SizeOfImage, SizeOfHeaders, CheckSum, Subsystem, DllCharacteristics, SizeOfStackReserve, SizeOfStackCommit, SizeOfHeapReserve, SizeOfHeapCommit, LoaderFlags, NumberOfRvaAndSizes, SectionsNb, SectionsMeanEntropy, SectionsMinEntropy, SectionsMaxEntropy, SectionsMeanRawsize, SectionsMinRawsize, SectionMaxRawsize, SectionsMeanVirtualsize, SectionsMinVirtualsize, SectionMaxVirtualsize, ImportsNbDLL, ImportsNb, ImportsNbOrdinal, ExportNb, ResourcesNb, ResourcesMeanEntropy, ResourcesMinEntropy, ResourcesMaxEntropy, ResourcesMeanSize, ResourcesMinSize, ResourcesMaxSize, LoadConfigurationSize, VersionInformationSize

in the total 56 features were extracted.

b- Data Cleaning:

the data set contained an attribute called Unnamed: 57 full of NaN instances , was drop later.

c- Feature Selection:

Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features). These new reduced set of features should then be able to summarize most of the information contained in the original set of features.

For that purpose, the selection was based on the Extra Trees Classifier, The Trees classifier is based on an extraction of observations from the dataset and an extraction of attributes.

From the 56 features, 13 relevant were selected and sorted according to their importance.

```
1. feature Characteristics (0.177768)
2. feature DllCharacteristics (0.122191)
3. feature MajorSubsystemVersion (0.078124)
4. feature ResourcesMaxEntropy (0.057559)
5. feature SectionsMaxEntropy (0.056962)
6. feature Subsystem (0.054413)
7. feature ResourcesMinEntropy (0.049921)
8. feature VersionInformationSize (0.043653)
9. feature ImageBase (0.039657)
10. feature SizeOfOptionalHeader (0.029637)
11. feature MajorOperatingSystemVersion (0.025945)
12. feature SectionsMeanEntropy (0.021445)
13. feature SectionsMinEntropy (0.019638)
```

Figure 3: Important Features

d- Data Standardization:

Standardize features by removing the mean and scaling to unit variance.

The standard score of a sample x is calculated as:

$$z = (x - u) / s$$

where u is the mean of the training samples or zero if **with_mean=False**, and s is the standard deviation of the training samples or one if **with_std=False.**

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using transform.

4.Approach Performance

As mentioned, the performance of each algorithm in classifying malwares is based on the accuracy, detection rate and rate of false positives and false negatives.

The accuracy, rates of false positives and false negatives are calculated using the confusion matrix. Confusion matrix, also called error matrix, is a table that shows different predictions and test results and compares them with the actual values. These matrices are used in statistics, data mining, machine learning models and other artificial intelligence applications.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 4: General Format of The Confusion Matrix

Accuracy: Accuracy is the simplest. It defines your total number of true predictions in total dataset. It is represented by the equation of true positive and true negative examples divided by true positive, false positive, true negative and false negative examples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Detection Rate(DR) : indicates the ratio of the number of instances that are 210 correctly classified as attack to the total number of attack instances present in test set.

$$DR = \frac{TP}{TP+FN}$$

False Alarm Rate: represents the ratio of instances which is categorized as 213 attack to the overall number of instances of normal behavior.

$$FAR = \frac{FP}{FP+TN}$$

Algorithms Performance The following table shows the results of the performance criteria values for each of the tested algorithms:

Algorithm	Accuracy %	Detection Rate%	False Alarm Rate%
Random Forest	98.4	97.72	1.23
KNN	97.42	96.66	2.17
SVM	94.42	95.15	3.33
Naïve Bayes	88.3	73.1	3.55

5.Discussion

From all the 56 features in the data only 13 features were relevant. This can be explained by looking at the nature of the malware binaries and the functionalities they are programmed to perform. For example , Characteristics is a relevant feature with an importance of over 0.17.

From the experimental results , we can notice that the Random Forest algorithm achieved the highest accuracy and the best in terms of in detection rate and false alarm rate, on other hand the Naïve Bayes achieved the worst results comparing to the other algorithms .

Jinrong Bai (2014) interpreted this by explaining that most types of malware are executable images, while most parts of benign software are dynamic link libraries. therefore, the mean values of characteristics, ImageBase, and Dllcharacteristics show significant differences between malware and benign software [7].

In many other studies, Random Forest did not achieve high accuracy while in this study the accuracy is higher. This can be explained by the

fact that the dataset used in this study is much larger (more than 216000 files), which makes this study outstanding.

6.Conclusion

Malware attacks are becoming one of the biggest threats to national information security, especially in these pandemic times when cyber-attacks are increasing tremendously. Malware programmers are developing new techniques every day to avoid detection. To counter these attacks, information security professionals must use new technologies to detect this malware.

In this paper , the method presented used PE headers to classify. The efficiency of these algorithms was studied in terms of accuracy, detection rate and false alarm rate. Random Forest showed high performance compared to the other 3 algorithms with accuracy above 98 percent.

REFERENCE:

1. Dragos¸ Gavrilut¸, Mihai Cimpoes¸u, Dan Anton, Liviu Ciortuz Malware Detection Using Machine Learning

2. Zane Markel and Michael Bilzor Building a Machine Learning Classifier for Malware Detection

3. P. Santhosh Raj, M.Phil Scholar Role of Data Mining in Cyber Security IJESC Volume 7 Issue No.7 2017

4. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges Journal of Network and Computer Applications · March 2020

5. dataset https://www.kaggle.com/competitions/malware-detection/data on March 13, 2022

6. https://docs.microsoft.com/en-us/windows/win32/debug/pe-format

7. Jinrong Bai, Junfeng Wang, and Guozhong Zou a Malware Detection Scheme Based on
Mining Format Information Hindawi Publishing Corporation The Scientific World Journal
Volume 2014, Article ID 260905, 11 pages