MALWARE DETECTION AND CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

Submitted in fulfillment of seminar required for the Master of Technology

Computer Science and Engineering

by

Abhishek Dhiman
173050066

Guided by
Prof. G Sivakumar

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra, India – 400076
Spring Semester 2018
Acknowledgments

I would like to express my special thanks and gratitude to my guide Prof. G Sivakumar for giving this golden opportunity to do this seminar on the topic "MALWARE DETECTION AND CLASSIFICATION USING MACHINE LEARNING TECHNIQUES" and also for guiding me by suggestions and constructive criticisms which also helped me in getting a good knowledge about machine learning and Computer Security.

Abhishek Dhiman (173050066)
Abstract

Malware is very dangerous in today’s world for that internet users. Nowadays, malware designed by attackers are generally polymorphic in nature. Polymorphic malware is a type of malware that constantly changes its identifiable features in order to fool detection using typical signature-based models[4]. Behavior-based malware detection evaluates not just on the signature of the file but also based on the action it wants to intend that is also before it actually executes that behavior. We want to obtain the behavioral pattern which can be obtained from static analysis or dynamic analysis, after that, we can apply different machine learning models in order to detect whether its a malware or not, or classify it to know malware families[5]. In this report, I will discuss Behavior-Based Detection methods and how we can apply different machine learning techniques in order to build behavior-based malware detection and classification methods.

Keywords: Static Analysis; Dynamic Analysis; Machine Learning; Classification; Behavior based detection methods
Contents

Acknowledgements

1 Introduction

2 Related work

2.1 Different types of Malware

2.2 Malware analysis technique

3 Methodology

3.1 Using Concepts of machine learning and data mining for Detecting and Classifying Malwares

3.1.1 Selecting the right features

3.1.2 Different classification Algorithms

3.2 Malware classification and detection using image processing method

3.2.1 Conversion of Malware binary to image

3.2.2 Extraction of Texture Feature

3.2.3 Classification technique

3.3 Popular machine learning techniques for malware detection based on their behavior

3.3.1 Data collection and report generation with automatic method

3.3.2 Data Preprocessing and Feature selection

3.3.3 Different Classification algorithms used

4 Results and Future Work

4.1 Performance Metrics

4.1.1 Approach 1

4.1.2 Approach 2

4.1.3 Approach 3

4.2 Results

5 Conclusion

References
List of Figures

3.1 Figure shows How to convert binary file to the image representation . 8
3.2 Figure shows how an image can be used to create textual features. . . 9
3.3 KNN example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
3.4 General overview of the methodology. . . . . . . . . . . . . . . . . . . 12
Chapter 1

Introduction

We all know the importance of the internet in our life’s. It has grown rapidly in the recent decades. With this trend, there are also a large number of hackers and terrorist those having an intent of doing crimes are creating malware[1]. Also, with a large number of tools available nowadays the amount of skills required to create a new malware is decreasing rapidly. Most of this malware are created for earning illegal money. As one of the report[2] 24.3 billion US Dollar’s are lost in the system of e-payment and a huge number of systems are infected because of keyloggers.

In 2017 internet security companies define malware as “a type of computer program designed to infect a legitimate user’s computer and inflict harm on it in multiple ways”. Also according to one another test, they register every day around 390,000 new samples of malware, with this large number processing process the data after doing the malware analysis is a difficult problem [3]. According to all above statements, it is clear that how difficult it is for anti-malware companies to tackle attacks and also releases there new updates within a limited time to prevent their customers from malware infection.

It must we clear Malware protection is a very important task of anti-malware companies as we know a huge data and money can be lost just because of one single attack. We can this fight a race between malware writers and malware protectors. So far most of the malware protection strategies rely on automatic malware analysis using automatic tools[6]. And these are basically signature or pattern-based methods. But the main point is now days malware writers have techniques such as polymorphism which can fool these pattern-based methods very easily. But there is a different approach that can try that is malware detection behavior-based techniques[8]. This approach is very different than pattern based as here check behavior and collect its information by executing it and that information we will use inorder to detect malware or not[9]. In general, most of the time we can observe that new malware’s that arise on a regular basis are just a little changed version of older malware using some clever techniques. So, this should be clear by now that techniques based on behaviour of malware are a great choice in order classify or detect malware.
Chapter 2

Related work

In this chapter i will describe some background that will useful in order to understand some of the techniques i will explain latter. Firstly, i will discuss malware types and then type of malware analysis. After that, I will discuss some machine learning techniques used by different authors for malware detection.

2.1 Different types of Malware

Malware can be classified into different categories based on how they try to infect or based on there behavior’s[5]. List is follows :

- **Trojan**: “A malicious application which present itself as something else[5]”.
- **Virus**: “Software which infects other application and uses them as a spreading medium[5]”.
- **Rookit**: “Hidden tools providing stealth services to its writer[5]”.
- **Worm**: “Code with ability to spread from computer to computer by means of different network protocols[5]”.
- **Spyware**: “Application aiming to harvest personal information[5]”.

2.2 Malware analysis technique

I describe in the above section that malware can be detected based on there pattern’s or how they behave. Let’s first discuss some techniques for analysis for malware.[8].

- **Analysis using dynamic methods** In Dynamic methods for malware analysis, we execute the file and collect the information about its properties like what actually is its intend[8]. What we can do is we can run our file by creating a virtual system such as Virtualbox while doing an analysis of this type, we can easily figure out all the behavioral based attributes example detecting a file, undo the file etc[8]. We can define behavior-based methods as both static and dynamic
analysis, whereas signature-based techniques are static most of the time. The advantage these dynamic methods is that we can make sure that what actually will happen when this type of malware’s in an actual system.

- **Analysis using static methods** A static method of analysis of malware is like using the fix patterns on source code and detect is behavioral properties[7]. Analysis using static methods have an advantage that with good accuracy they can detect its purpose and functionality[3]. K. Rieck et. al. 2008 have discussed some of the popular types of static analysis techniques for malware detection:

1. **Inspection on File Format**: The format of the file or metadata can be a good way to guess what is actually the intended action. For example, Portable executable files in Microsoft Windows is good way to know about information like executable or compile time and other functionalities.

2. **Extraction of Strings**: In this process, we can guess the information about the malware and its operation by analyzing the output of the software example it can be a status or an error message.

3. **Fingerprinting technique**: In this method, we can do sophisticated things such as computing cryptographic hash[3] and figure out things like username, the name of the file, string related to the registry which is hardcoded.

4. **Scanning with Anti-Virus**: We all use Anti-Viruses, it can easily detect malware’s which are well-known. So, this technique can also be used by sandboxes (virtual machines) in order to detect malware.

5. **Disassembly method**: This one is a very common technique for Analysis using static methods. In this technique, we will convert our code to assembly code and use this we will guess what is actually is intended.

M. Egele et al.2007 have used Analysis using dynamic methods for the first time and are able to get the information about intending of some malicious codes reliable and accurately. Also, M. Bailey et al.2001 is basically based on trying to automate processes involved in Analysis using dynamic methods one of them is information extraction. M. Egele et al.2001 discussed techniques based on Analysis using behavior methods using clustering methods which is an unsupervised machine learning methods. In the technique discussed by M. Egele et al.2001 we basically can transform the behavior data we observed into a sequence and for the measurement of distance in clustering we can use this distance and we can group them into family of malware clusters. But there are also many difficulties involved in clustering methods because of there unsupervise nature that is for the analysis of data we have no external details or information. One of the major problem [6] is how many cluster are there in the data is difficult and require some domain knowledge.
Chapter 3
Methodology

If we see literature of detection malware methods based on machine learning methods different models are those are successfully used are Su

Various machine learning methods are found useful in the literature of malware detection and classification some of these techniques are Support vector machines, ensemble models like random forest, naive Bayes and also we can group together different malware families based on their hidden behavior using the unsupervised techniques (e.g. clustering like K-means), but major challenge is to decide number of clusters or groups. In this section, I will discuss few papers those have used this technique for detection and classification of malware families.

3.1 Using Concepts of machine learning and data mining for Detecting and Classifying Malwares

In paper M. Schultz et al. 2001 firstly discussed this idea of detecting the malware with machine learning and data mining techniques. With the experiment [8] it can be shown that we detect and classify malware accurately and automatically using our data mining and machine learning techniques. Results of paper by M. Schultz et al. 2001 was much better than old pattern-based detection methods,

3.1.1 Selecting the right features

In paper M. Schultz et al. 2001[8] used a dataset which contains around 4,299 programs out of these many programs they split into a malicious program of size 3100 and remaining as the clean programs. All the program in the dataset was labeled. Labeling was done with help of an anti-virus scanner. Labeling was done for either malicious or benign. M. Schultz et.al 2001 have also three types of static feature for training machine learning models in order to detect malware and classifying them (the main focus was on classification). These three types are as:

1. Portable Executable (PE)
In order to extract information in the format of the object from the portable executable header, a library within Bin-Utils that was libBFD. Some of the features we obtained from here are the size of the file, the names of Dynamically linked libraries and Dynamically linked libraries function calls. With the Portable Executable approach, some other features are "list of DLLs used by the binary[8]" and also the count system calls within each Dynamically linked libraries are extract is used as a feature.

2. String Sequences

Features using the string are also extracted from files depending on how these strings are encoded inside these files. Based on their experiment M. Schultz et al 2001 found that string pattern those are in clean programs similar in all clean files and this makes them different from the malware files also it’s in the other way malware files have different patterns that make them from clean files. This method is not very different from traditional pattern-based detection methods but different here is we are using this step of features selection. The major issue with these type of features they lack in term of robustness as using clever techniques they can easily be changed, that is why another type of features can be used called byte sequences.

3. Sequence of Bytes

Byte sequence method for selecting or extracting features used the n-gram based technique on executable files. Using n-gram and a tool called “hexdump” hexadecimal files can be obtained from the binary files. If we compare these features to another type of feature author of paper M. Schultz et al.2001 found that Sequence of bytes is most useful as it has machine code executable as compare to resource information such as portable executable features. If we want to see second most important feature then we can say that program executable is more useful then string features as they are not robust.

3.1.2 Different classification Algorithms

In this paper [12], they used three type of algorithms Ripper algorithm, Multinomial naive Bayes and naive base for this system. Let’s discuss them one by one:

1. Ripper algorithm

Ripper algorithm was very similar to the pattern-based algorithm similar to that goal here was to find the pattern in the string data. The resource rule built in the string data and later we can use them for future for detecting malware.

2. Naive Bayes

Naive Bayes is a classic machine learning algorithm in which we can use all our feature to detect whether they become malicious file or not and used it for the purpose of classification. For our purpose, we can use it to find the likelihood of
being malware given all the features.

\[ P(C|F) = \frac{P(F|C) \times P(C)}{P(F)} \] \tag{1}

The first assumption of naive Bayes is all features occur independently (Definition of Naive Bayes). If we have the feature of the program W include W1, W2, W3 to Wn. we can write this equation as

\[ P(C|W) = \Pi_{i=1}^{n} \frac{P(W_i|C) \times P(C)}{P(W)} \] \tag{2}

Here, each P(W—C) is nothing but how frequent the string W in a program of class C (Because classification is our major task here) in the entire set of programs. Therefore, the output represents the family which is highly probable based on the given features. As we can see the part which is in the denominator of (1) will be same throughout each family or class. We will just take the maximum probability class from (2) in order to get:

\[ MaxLikelyClass = \max_{C}(P(C)\Pi_{i=1}^{n}P(F_i|C)) \] \tag{3}

In (3) author for the paper used such notation as we will take a class of family with the highest probability. Hence, Output we get is very simple that is most likely class given all the features. So, in order to train the classifier author of paper M. Schultz et al.2001 count the number of files in each class that have unique features. Soo, basically we can use that information our new file to an appropriate family of class. I can some up the algorithm by saying that in the first step we extract features in order to see if features that contained in given file and then we apply equation (3) to get the most likely family it belongs to.

3. Multi-Naive Bayes

Multi-Naive Bayes method is nothing but an advanced version of naive Bayes with voting methods. We apply the different naive based method and later we combine them by taking the output from one of them and doing a voting on them. The final output is given by Multi-Naive Bayes algorithm which combines all the Naive-Bayes algorithms.

Note, that not all the feature are given when using these are multi-naive Bayes but only some feature to one model and soon. This is space efficient method because with such a big feature size the size of RAM we need is very large to fit such model. With such a big RAM requirement completing this task is very difficult but here the table we require for all features in order to find probabilities table is divided into pieces and different naive models are trained with them. But here important thing is to divide data evenly[8]. Soo, the jth binary line is put
into the \((j \mod n)\)th set. Here author used set size equals to 6. So, basically, we can train these 6 models and combine them with voting on them.

We can see this problem as our model is combining the vote of confidence by combining different individual naive classifiers. Our individual model gives us a probability of a class \(C\) given a set of features \(W\) and our Multi-naive ayes classifier us it to compute the probability of that class given all the features. The goal is to compute the likelihood of a given family or class given features \(W\). In the equation below we are computing this

\[
L_{NB} = \prod_{i=1}^{NB} \frac{P_{NB}(C|W)}{P_{NB}(C)}
\]

where \(NBi\) is a Naive Bayes classifier and \(NB\) is the set of all combined Naive Bayes classifiers. So, basically our multi-classifier based on the feature \(W\) gave us the family of highest probability over all families given \(L_{NB}(C|F)\) and \(P_{NB}(C)\)

\[
\text{MaxLikelyClass} = \max_{C}(P_{NB}(C) \ast L_{NB}(C|F))
\]

Therefore, a most likely family of malware among all families in \(C\) with the highest probability with features \(W\), and \(\max_{C}\) returns the highest with the highest likelihood.

The result of the paper M. Schultz et al 2001. I will discuss in the results section. Results are later improved in the next of next paper by authors of Kolter et al 2004. Instead of using the non-overlapping byte sequence as we used there we[10] can use n-gram technique. So, basically instead of using naive-Bayes we are doing some wrong assumption instead models like Random forest, Support Vector Machine outperforms naive Bayes method. Authors of Kolter et al 2004 shows random forest can give best classification results.

### 3.2 Malware classification and detection using image processing method

Different machine-learning methods we saw in the previous section can be used for malware detection problem it can be supervised or unsupervised learning method. Author of paper Nataraj et al .2011[14] gave an out of box algorithm for this problem. Problem with an earlier type of models is that our time of feature extraction does not scale with malware files or input binaries. The approach I am going to discuss here is also called as binary-texture analysis[14].

The technique discussed in the paper Nataraj et al .2011[14] are widely used in different domains like speech recognition, medical imaging, and other areas of machine learning applications. So, basically the process is a very simple binary of the file is first converted to an image and using this image representation of the binary the
texture based features are obtained. Author of the paper Nataraj et al. 2011[14] use this technique and compared it with earlier approach I discussed. Let’s discuss in detail

### 3.2.1 Conversion of Malware binary to image

The first step in this approach is to convert binary files it can be malware or not to an image so that we can extract image-based features. In step of converting binary to image our binary file is converted into a 1D array that is vector form of 8-bit unsigned integers[14]. After performing the step one this array or vector is organized as a 2D array or matrix. For matrix created width and height depend on the file size we used or binary input[14]. Figure 3.1 shows How to convert binary file to the image representation and Figure 3.2 shows how an image can be used to create textual features.

![Figure 3.1: Figure shows How to convert binary file to the image representation](image)

### 3.2.2 Extraction of Texture Feature

In the previous, we end up computing an image from the binary file now we can obtain our features which can help us in building machine learning model. The author of the paper Nataraj et al. 2011[14] used an image processing algorithm called GIST which is very popular in other domains also. GIST works pretty well other options are pyramid histogram of oriented gradients (PHOG) for the same purpose and also we can use state of the art Deep learning methods which I think will be more successful then both GIST and PHOG. Figure 3.2 shows the block diagram to obtain the GIST feature. The steps to compute the texture feature are as follows:

1. Binary bytes are converted to 1D array or vector(range 0-255).
2. Use above vector and convert to 2D array in order to get the grayscale image. And keep the width constant based on file size of the binary[17].

3. Change the image to a 64*64 square image.

4. Use N=20 sub-bands[14] in order to 16-dim text features.

5. Get 320-dim feature vector by combining all the features.

3.2.3 Classification technique

If we see the previous literature on the classification of image categories using Texture-based image features one of the best features is KNN. One example is given by the author of paper L. Nataraj et al. 2011[16] classify outdoor scene categories. Let’s discuss the complete approach in detail

**K-Nearest neighbors** K-Nearest Neighbors or KNN is very simple and accurate machine learning model[16]. It contains no parameters in the sense that it no assumptions about the data-set. If we talk about some real problems if in that problem data obeys assumptions, then only making a non-parametric assumption is a good solution. K-Nearest Neighbors or KNN model representation is as simple as the dataset and it involves no learning of parameters.

K-Nearest Neighbors or KNN can be used for both classification or regression problems. And in both cases, we output based on those k training examples or instance which are closer to our input, which is like majority voting I discuss above with distance measure that needs to define. Similarly, in case of regression, we will take an average of all the k examples. Example in Figure 3.4 Let’s see how to apply these techniques to our problem. The author used labeled data in training set that is known malware families. Each family of malware has variants of malware. As explained in above section GIST feature’s of 320 dimensions are extracted from the training set[14]. Let’s see each step in the approach

1. From the unknown malware binary GIST features of 320-dimensions are obtained.
2. In a second step, we calculate and store the Euclidean distance with all feature vectors or sample of the training set.

3. After doing above step we find the k closest malware families in terms of Euclidian distance.

4. Final to this unknown malware we assign a class that is in majority among these k neighbours.

The limitation is that popular adversarial attacks are possible against these types of approaches. This method uses global image-based features which a hacker can adopt countermeasures to beat the system. An attacker can do many things to corrupt binary to chance the gray-scaled image. The result of the paper I will discuss in the next chapter.

3.3 Popular machine learning techniques for malware detection based on their behavior

You have I have described different machine learning technique and feature extraction methods for dealing with the problem of malware detection with machine learning. Firdausi et al. 2010[18] tried various different such approaches and summaries them experimentally in a very nice manner. They used emulated(sandbox) environment in order to analyze the malware and getting the report of its behavior automatically. Let’s discuss this complete approach step by step.
3.3.1 Data collection and report generation with automatic method

Let’s first discuss the dataset, it contains both the malware and clean files[18]. And more importantly, these instances of data sets are in the binary file format of Windows. A total of around 220 unique malware samples were collected. Also, they collect clean system file from a clean installation from “system files of Windows XP Professional”.

The Key step is to generate the Report Generation. The report is generated by conducting behavior monitoring[18] with our both classes that are malware files and clean files.

3.3.2 Data Preprocessing and Feature selection

Data Preprocessing was the key step in their research[18]. I break down them in following steps.

- For feature select XML report files that we obtained from the previous section is convert to get most relevant and important attribute values.

- After selecting the necessary attributes a data structure is created that store attributes which is previously selected[18]

- This data structure is used to compare with XML report file and we count the existence of each word in the data-structure binary weight and words frequency weight[18].

- Two types of files are created a sparse vector model for each report and also Attribute-Relation File Format(ARFF) files for them[18].

3.3.3 Different Classification algorithms used

Now we have feature’s that is ARFF files the step is to try different machine learning algorithms on top of that. and after building ARFF files we will apply models on top of that[18]. Before applying models they divide the dataset into following parts:

1. Without feature selection apply Binary-weight vector model.

2. without feature selection Term frequency-weight vector model.


4. feature selection Term frequency-weight vector model.

Now on each data set we can apply different types of models. The models applied by the author of paper Firdausi et al .2010 [20] are Multilayer Perceptron (MLP), SVM and Decision tree’s. Results i will discuss in next chapter. Figure 3.4 describe the overview I discussed above.
Figure 3.4: General overview of the methodology.
Chapter 4

Results and Future Work

In this report I explained three techniques for malware detection and classification. First approach used data mining and machine learning algorithms like k-Nearest Neighbor, Nave Bayes, Support Vector Machine (SVM), J48 decision tree, and Multilayer Perceptron (MLP) neural network. Second approach was using image recognition algorithms and in third approach I discussed the analysis of machine learning algorithm for malware detection and classification.

4.1 Performance Metrics

4.1.1 Approach 1

For the first approach we can measure the performance based on three measure’s true positives (Data points those are actually true and are classified true), false positives (Data points those are actually false but are are classified true) and finally we can compare accuracy of different models that we discussed in the first section [12].

4.1.2 Approach 2

For the second approach which is a type of multiclass classification we can use accuracy, confusion matrix for 63,002 malware comprising 531 families for data-set one, and confusion matrix for 500,000 malware comprising 831 families for data-set two [14].

4.1.3 Approach 3

For the third approach we can measure performance statistically, based on malicious or benign tests. These statistical measures include positive predictive value (precision), true positive rate (sensitivity, recall, hit rate), false positive rate (fall-out), and accuracy (Number of truly classified points divided by total number of points.) [18].
4.2 Results

In the approach 1 based on the false positive rate was 2.13% which is very good as we compare to earlier results. And based on the true positive rate they were getting 98.1% results. And finally based on accuracy they got 98% results.

For approach two result is little complicated firstly for dataset one that is VX-Heavens[19]. Average accuracy obtained is 0.728%. There were around 531 families with accuracy 1 around 33 families of malware are classified. Whereas 105 families are getting classified with an accuracy above 92%. Where for data-set 2 that is Anubis Data-set[19]. Initially, there was no labeling on the data-set so clustering is applied to get the similar malware families grouped together. After doing this operation we can apply can classifier which is supervised in nature using cross-validation and accuracy was 0.718.

For approach three which was a deep analysis of different machine learning based on 3 metric’s L. Nataraj et al .2011 compared them, Here is the summary of this analysis.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>TPR</th>
<th>FPR</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>81.7%</td>
<td>8.1%</td>
<td>86.5%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>58.1%</td>
<td>9.8%</td>
<td>65.4%</td>
</tr>
<tr>
<td>SVM</td>
<td>90.4%</td>
<td>8.4%</td>
<td>91.0%</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>90.9%</td>
<td>3.8%</td>
<td>93.6%</td>
</tr>
</tbody>
</table>

We saw that in L. Nataraj, et al .2011 paper we extract all feature using old image featuring extracting methods. We can replace this technique by state of the art in most of the image application deep learning methods to get better accuracy.
Chapter 5

Conclusion

I would like to conclude by reporting how further research is needed in this area of Malware detection and classification since internet is reaching more and more people every day. Also, malware creation is becoming simple data by day and I showed using three different techniques that machine learning can be of great help in defending us from that.
References

[1] Radu S Pirscoveanu, S Steven, Thor MT Larsen, Matija Stevanovic and Alexandre Czech, “Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)”, Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), International Conference on 2015.


