Innovations in Computing Technology and Applications
Volume 2 | Year: 2017 | ISBN: 978-967-0194-95-0

Classification of Malware Family Using Decision Tree Algorithm

[1]Mohd Shaiful Anuar Bin Mohamad Sari, [2]Mohd Aizaini Maarof

Department of Computer Science, Universiti Teknologi Malaysia, Skudai, Johor

[1]saifulanuar2030.[2]sa@gmail.com

Abstract—Malware detection and classification is important in providing best security for computer systems. Current detection and classifying techniques cannot give the best and accurate in classifying the malware while dealing with new types of malware. With the rapid increase of malware types, accuracy is important to classify unknown malware into its own family. Currently, the majority of antivirus detection system are signature-based, which means that they try to identify malware based on single featured. Cannot detect unknown malware but only identify variants malware that have been previously identified is the disadvantage of signature-based detection system. Therefore, an improve decision tree algorithm is used to classify malware correctly. The result of accuracy on classifying into its family are 93.3% on multiclass and 94.6% on binary class. This result is higher than others machine learning being tested.
Keywords- *malware; behaviour; visualization; classifying; decision tree*

1. Introduction

Malware has already become one of the biggest problem which has affected different parts of the world in different ways. From its very beginning in the 1960s, malware have developed into the most significant threat to computer system, especially in the last three decades. There has been an intense evolution in occurrences of malware, along with the growth of the internet in recent years.

With the rise of the shadow Internet economy, malware is no longer simply used to damage, break or interrupt on computer system, but now occur primarily as a tool used by illegitimate person to make a profit. Malware makers are often looking for one-time development of specific code to generate new variants of existing malware, instead of developed new malware from scratch. In this case, variants of existing malware can be developed easily and quickly, and therefore, can be rapidly brought to market in the shadow economy.

Thus, there is need to develop an integrated classification technique to classify the variants of existing malware, in order to guide analysts in the selection of samples that require the most attention. Over the last decade, researchers have adopted a variety of solution to control malware.

Furthermore, visualization also needed to bring awareness to people about the malware and its behavior. With a vast number of samples increasing daily, it is harder for anti-virus industry and virus researcher to analyze malware without information of new malware. It is necessary to have an automatic malware classification system, in order to reduce time of malware analysis.

The objectives of this research are: (1) to identify the characteristic and structure of the existing malware from file's meta-data, (2) to classify the malware by using decision tree technique, and (3) To visualize or develop a dashboard using the data that had been analyzed.

2. Method

The research framework comprises of three phase comply with all three objectives in chapter 1. This work consists of various step to construct it. Many processes are required in this approach such as data planning, pre-processing, characteristic identification, classify using decision tree algorithm and visualization.

2.1. Phase 1: Pre-processing the Data

Phase 1 of the project consist of three stages which the main objectives is to identify the characteristic and structure of the existing malware from file's meta-data. All the sample data will be identified its own behavior. All the sample will be test using Cuckoo Sandbox. The Cuckoo will generate report that will determine the behavior of the malware. All the sample will be store using csv format file.

It is important to configure Cuckoo Sandbox to make sure we get the malware behavioral reports and also to ensure malware run correctly, including all of its functionality. It is also important to include a broad range of services in the virtual machines created by the sandbox as in the real world, different vulnerabilities that might be part of certain software products can be exploit by different types of malware sample.

For Cuckoo, Virtualbox as the hypervisor is used for the virtual machine. VMcloak will be used to create virtual machine. As stated by [3], VMcloak is an automated virtual machine generation and cloaking tool for Cuckoo Sandbox.

The following specification will be used at all virtual machine:

i. 1 CPU core 3.2 Ghz
ii. 2 GB RAM
iii. Internet connection

On all the virtual machines, this following software will be installed:

i. Windows 7 Professional 64bit without any updates, including Service Pack 1
ii. Adobe PDF reader 9.0
iii. Adobe Flashplayer 11.7.700.169
iv. Visual Studio redistributable packages 2005 – 2013
v. Java JRE 7
vi. NET framework 4.0

2.2. Phase : Features Identification and Classification.

Phase 2 of the research consist of three stages which the main objective is to classify malware into each malware family using the decision tree algorithm. The sample then will be used for training and testing using Decision Tree algorithm.

Feature extraction that will be used is the combining matrix that includes successful APIs, failed APIs and their return codes. From the reports generated by the sandbox, the data is extracted.

Figure 1 will explain the detailed process of the feature extraction. The report generated by the sandbox are used as an input to the feature extraction which then will produces the .csv file.

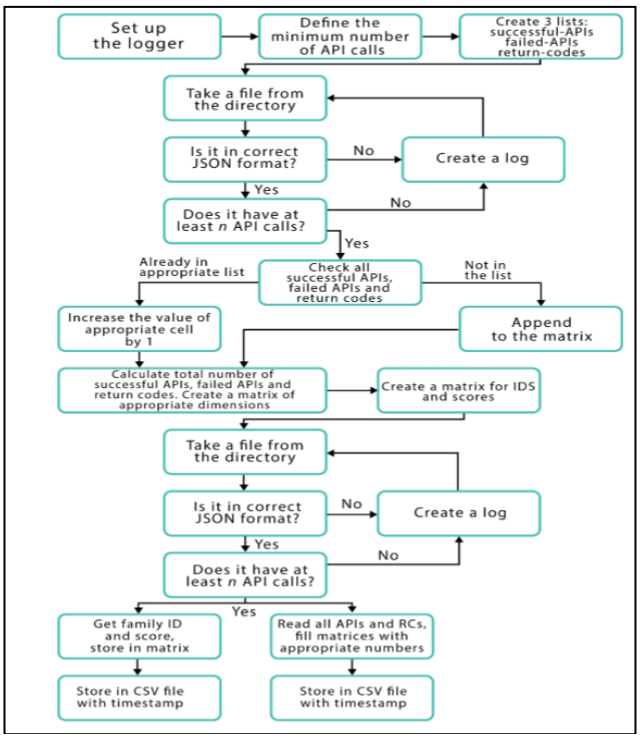


Figure 1. Feature Extraction Process

For performing the feature selection and applying the machines learning methods, the language R will be used. For statistical computing and graphics, R is a free software for that environment. According to [11] R language compiles and runs on wide variety of UNIX platforms, Windows and MacOS.

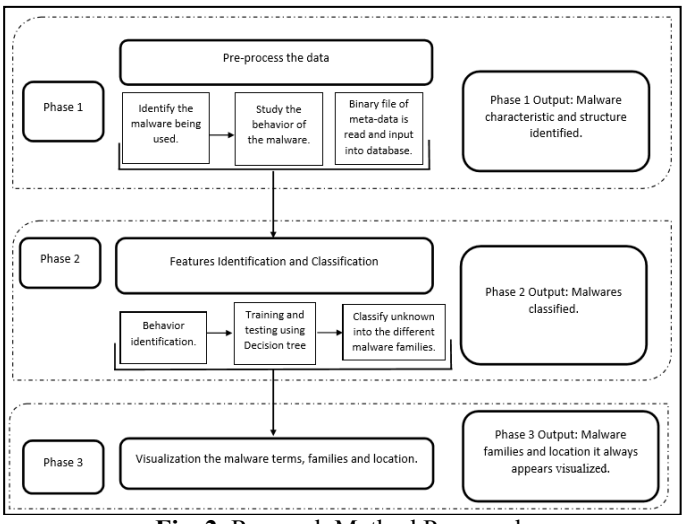
The Boruta package is a good and simple algorithm for feature selection in classification problem. As stated by [4], it is a wrapper method that works around the Random Forest algorithm. Follows are the description about the algorithm [6]:

i. Create shuffled copies of all features (to add more randomness). These are referred to as shadow copies
ii. Train a Random Forest classifier on the new dataset and apply a feature importance measure in the form of the Mean Decrease Accuracy algorithm. The importance of each feature is measured at this stage, and the weights are assigned
iii. On each iteration check if the feature from the initial feature set has a higher weight than the highest weight of this feature's shadow copy. Remove the features that are ranked as unimportant at each iteration.
iv. Stop after classifying all features as 'selected' or 'rejected', or after a certain number of iterations of random forest is achieved

2.3. Phase 3: Visualization

The result is visualized in the form of a simple dashboard using Tableau public software. The data is analyzed by the software to show the type of malware attack in several countries.

After all the data, already being classified accordingly into its families, it then will be visualized using visualization tools call Tableau. Tableau is a software that can extract data from multiple source such as excel, MySQL and etc... It also can extract data even it is in .csv format which is the output of these experiment.



Fig, 2. Research Method Proposed

3. Result

In this research, we perform malware sample classification using Decision Tree Algorithm. By using the percentage of correctly identified instances, the accuracy of detection is measured.

$$Accuracy = \frac{count(Correctly\ Identified\ samples)}{count(Total\ samples)}$$

The overall accuracy of decision tree was 93.3% for multiclass while for the binary classification, the accuracy was 94.6%. The result can be seen in the table below.

3.1. Multiclass Accuracy

Table 1. Multiclass Accuracy

FAMILY	CORRECTLY CLASSIFIED	INCORRECTLY CLASSIFIED	ACCURACY
BENIGN	54	7	88.5%
Win32/Virut	37	0	100%
Win32/Autorun	24	3	88.9%
Win32/IRCbot	44	0	100%
Win32/Gaobot	16	2	88.9%
Win32/Waledac	33	7	82.5%
Win32/Downadup	47	2	95.9%
Win32/Sality	38	0	100%
Win32.Mota	32	2	94.1%
Locky	21	1	95.5%

For multiclass accuracy, it calculates all the accuracy of the malware sample and divide it according to it family.

3.2. Binary Accuracy

The algorithm, in binary classification, it divided only two type which is benign and malware. It resulted in 46 correctly identified instances for benign sample (true negatives), 305 correctly identified for malware sample (true positive), 15 incorrectly identified for benign sample (false positive) and 5 incorrectly classified for the malware sample (false negative). Tables 2 and 3 will present the detailed result.

Table 2. Decision Tree Binary Classification accuracy

CLASS	Correctly Identified instances	Incorrecly identified instances	accuracy
Benign	46	15	75.4%
Malicious	305	5	98.4%

Table 3. Decision Tree Binary Class Accuracy

TRUE POSITIVE	TRUE NEGATIVE	FALSE POSITIVE	FALSE NEGATIVE
305	46	15	5

93.3% for multiclass and 94.6% for binary classification was a good result for accuracy of Decision Tree in classified the malware. As for these result, it was much better compare to the other machine learning technique. Hence, this is the proof that Decision Tree is better than other machine learning technique.

3.3. Visualization

After all the classification has been finished. The result then will be presented and visualize using Tableau software. This visualization includes type of malware and family, its behavior and common places that the malware being reported.

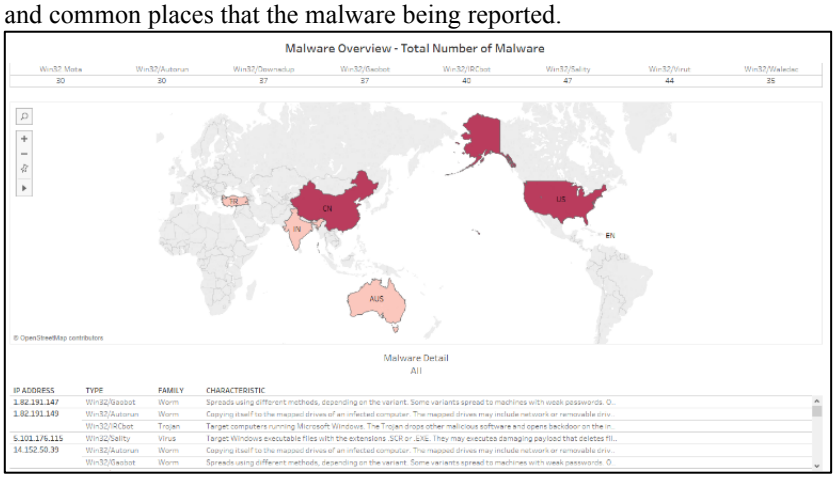


Figure 2. Visualization of Malware Characteristic, Family and Common Places Being Reported

This visualization can help people to increase their awareness about the characteristic of the malware. This also can help them to make a good decision in taking an action if their

devices being infected by certain malware as they can determine what are the type of malware infected them. Then, a proper solution can be taken to neutralize the malware. This can save their time to take the right action.

3.4. Comparison between Decision Tree result and other machine learning

The result obtain then be compared with other machine learning such as K-Nearest Neighbor, Support Vector Machine and Naïve Bayes.

Table 4. Naïve Bayes Binary Classification Accuracy

Class	Correctly identified sample	Incorrectly identifies sample	Accuracy
Benign	61	0	100%
Malicious	143	167	46.1%

Table 5. Naïve Bayes Binary Classification Accuracy

True positive	True negative	False positive	False negative
143	61	0	167

Overall, performance of Naïve Bayes is not good enough. The percentage of binary accuracy is only 55%. For real world detection, this result is insusceptible. Moreover, the number of false negative, which means that it incorrectly classified malicious file as benign file, is 167. It is almost half of the total sample. Such result, in real world, would cause a huge malware increases in short amount of time.

Table 6. KNN Binary Classification Accuracy

Class	Correctly identified sample	Incorrectly identifies sample	Accuracy
Benign	49	12	80.3%
Malicious	302	8	97.4%

Table 7. KNN Binary Classification Accuracy

True positive	True negative	False positive	False negative
302	49	12	8

Overall, for KNN classifier, the accuracy percentage is high but the false negative is higher than Decision Tree. It means that Decision Tree is more accurate to classified malware than KNN. Even though the accuracy percentage are same between this two, Decision Tree is better to classified malware correctly and it can be seen at the false negative.

Table 8. SVM Binary Classification Accuracy

Class	Correctly identified sample	Incorrectly identifies sample	Accuracy
Benign	41	20	67.2%
Malicious	310	0	100%

Table 9. SVM Binary Classification Accuracy

True positive	True negative	False positive	False negative
310	41	20	0

Overall, for SVM classification accuracy, the result is 94.6%. Even though the percentage of accuracy is similar to Decision Tree, the total of incorrectly identified sample of benign malware is high. It means that SVM identified certain benign malware as malicious malware. Also, as false negative is 0, it means that no malware sample was detected as benign.

4. Discussion

Overall, for this research, the goal defined were achieved. The selected machine learning, which is decision tree, was applied and evaluated along with the desired extraction and representation methods.

The combined matrix, which was the selected feature representation, outlining the frequency of successful and failed API calls along with return code. Because of the actual behavior of the file being outline by it, it was chosen as the desired feature representation. It combines information, different than other method, about different changes in the system, including the changes in the mutexes, files, registry, etc.

4.1. Achievement

In the classification problem, Decision Tree able to achieve highest accuracy with 93.3% for multiclass and 94.6% for binary classification. Also, the result achieved by Decision Tree is much better than the one achieved by the sandbox. Since the sandbox does not classified the sample into benign or malicious, it is hard to compare the result quantitively.

As we can see from the table above, the result achieve by Decision Tree was better compare to the other machine learning technique. Decision Tree can detect more accurately the samples being used for this research.

4.2. Limitation

There are a lot of obstacles we faced during this research period. Many alternatives were considered and carried out to fix the problem, due to this matter.

Throughout the entire research study, the problem that was difficult being faced was trying to get wider and up-to-date dataset. Collecting the dataset was a tedious task that require a lot of time and effort. Also, during preprocessing the data, it take too much time as we need to wait for the sandbox to generate report before we can continue with the next step.

Time was the biggest obstacle in this research. We sometimes take too much time to collect the sample of malware. Also after the sample had being collected, we spent more time afterward when we waited for the sandbox to generate the report which will be used in the classification.

4.3. Future Work

It is recommended to implement the classification based on the Decision Tree, based on the result achieved before, for multiclass classification, as it resulted as the highest accuracy. As for binary classification, it still need some improvement to achieve more accuracy as still detect some false negative, which was not so good.

There is some planning would be done in the future:

1.1.1 *Wider dataset need to be used.*

As we know, or the machine learning to be able to predict the malware, first it need to be familiar with the malware. So wider dataset will need to be used for the Decision Tree to train as it will make more accurate classification as it already familiar with the new or modern sample of malware. More dataset being used for Decision Tree to train, more accurate it can be classified the malware.

1.1.2 Try to combine with other machine learning method.

As we know, the Decision Tree may have high accuracy in the multiclass classification. But in the binary classification, there is another method that has higher accuracy than Decision Tree. It has 0 false negative which make it 100% accurate. So, we can be combined these two methods to make a hybrid classification technique so that we can achieve the highest accuracy without any false negative.

5. Conclusion

Even though the result achieved were pretty highly accurate, there still need some improvement in the future work to make it more accurate as in the real world, there are more families type of malware that exist. Overall, the result we achieved were good enough as we can point out which method was better to be implemented.

References

1. Bayer, U., Comparetti, P.M., Hlauschek, C., Kruegel, C. and Kirda, E. (2009). Scalable , Behavior-Based Malware Clustering. Sophia. 272(3), 51–88. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.148.7690&rep=rep1&type=pdf.
2. Bhawe, C. (2013). BIG DATA CLASSIFICATION USING DECISION TREES ON THE CLOUD. Available at: http://scholarworks.sjsu.edu/etd_projects.
3. Bremer, J. (2017). VMCloak Documentation.
4. Chumachenko, K. and Technology, I. (2017). Machine Learning Methods for Malware Detection.
5. Komashinskiy, D. and Kotenko, I. (2010). Malware Detection by Data Mining Techniques Based on Positionally Dependent Features. Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on., 617–623.
6. Kursa, M.B. and Rudnicki, W.R. (2010). Feature Selection with the Boruta Package. Journal Of Statistical Software. 36(11), 1–13. Available at: http://www.jstatsoft.org/v36/i11/paper.
7. Oost, N. (2008). Binary code analysis for application integration
8. Saxena, P. (2007). Static Binary Analysis And Transformation For Sandboxing Untrusted Plugins. (August), 48.
9. Schipka, M. (2007). The Online Shadow Economy: A Billion Dollar Market for Malware Writers. White Paper, MessageLabs Ltd. Available at: http://scholar.google.com/scholar?start=20&q=malware+market&hl=en&as_sdt =0,5#1.
10. Vigna, G. (2007). Static disassembly and code analysis. Malware Detection., 19–41.
11. Venables, W. N., and D. M. Smith. 2016. An Introduction to R.