Comparison of malware detection techniques using machine learning algorithm

Nur Syuhada Selamat, Fakariah Hani Mohd Ali
Faculty of Computer and Mathematical Science, University Technology MARA Malaysia Shah Alam, Malaysia

Article Info

Article history:

Received Nov 1, 2018
Revised Feb 6, 2019
Accepted Mar 15, 2019

Keywords:

Dynamic analysis
Hybrid analysis
Machine learning
Malware
Static analysis

ABSTRACT

Currently, the volume of malware grows faster each year and poses a thoughtful global security threat. The number of malware developed increases as computers became interconnected, at an alarming rate in the 1990s. This scenario resulted the increment of malware. It also caused many protections are built to fight the malware. Unfortunately, the current technology is no longer effective to handle more advanced malware. Malware authors have created them to become more difficult to be evaded from anti-virus detection. In the current research, Machine Learning (ML) algorithm techniques became more popular to the researchers to analyze malware detection. In this paper, researchers proposed a defense system which uses three ML algorithm techniques comparison and select them based on the high accuracy malware detection. The result indicates that Decision Tree algorithm is the best detection accuracy compares to others classifier with 99% and 0.021% False Positive Rate (FPR) on a relatively small dataset.

Corresponding Author:

Nur Syuhada Selamat,
Faculty of Computer and Mathematical Science,
University Technology MARA Malaysia 40200 Shah Alam,
Selangor, Malaysia.
Email: nursyuhadaselamat89@yahoo.com.my

1. INTRODUCTION

Today’s world is rapidly moving towards digitization. Computer field has gained a lot of importance in our daily life to deal with many aspects like business purpose, education etc. This scenario is very crucial for a country an organization in the context of protective and safeguarding the digital resources [1]. The word “Malware” stands for malicious software and it usually specifies as hostile software application [2]. Malicious software has developed into the most significant threat to computer system, from its very beginning in the 1960s.With the increasing of the internet users in recent years, there has been a powerful evolution in occurrences of malicious program. Along with the technology advancement, the malware authors have developed malicious code that hard and difficult to be analyzed and detected by researchers. For example, malware writers created malicious code with implement new technique mutation characteristic on that malware which causes an enormous growth in number of variation of malware.

Since the number of malicious software rapidly increasing [3], antivirus companies are continuously looking for a technique that is the most effective in detecting malware. Signature based detection is the most popular method used by antivirus company. However, the traditional malware detection strategies are not capable to notify the unknown malwares and only identify variants malware that have been previously identified.

Many efforts have already been made to detect malware. Several methods have been used in many research papers [4]. There are different types of malware detection and classification using techniques such as static, dynamic and hybrid features [1].Static analysis also called as code analysis [5] without execute malware by examining and observed software code to gain information of how malware' functions work. There is a completely different technique without using the codes but according to the runtime behavior by watching its behavior, system interaction and effects on host system called as dynamic analysis [6]. Whereas hybrid analysis [7] is a combination of static and dynamic analysis.

By using API calls, Tian et al. [8] proposed a binary feature method for malware detection and classification. In this paper, the writers also investigated the frequency based methods on the same data but no upgrading was observed over the binary representation. In similar approach, Z.Salehi et al [9] proposed a malware detection based on API calls and their arguments. The authors used this technique as a feature and analyzed their outcome on the classification process. To decrease the number of features feature selection algorithms are used. The result from the experimental evaluation shows an accuracy of 98.4% in the best case by using random forest algorithm.

D. Arshi and M. Singh [10] proposed a method based on behavioral analysis on machine learning that focused on classification and clustering of malware.In their experiment, they used two types of classifiers which are K-Means and Logic Model tree algorithms. The result showed that 82% aimed to corrupt in the computer system or network resources while 18% of analyzed malware were embedded with networking capabilities to connect the outer world. P.V.Shijo and A. Salim [11] proposed a method that provides the efficient automated classification of malwares by using both static and dynamic features of malwares and by using machine learning technique. In their experiment , the static features are extracted from the binary code while in dynamic analysis is done by using the tool cuckoo sandbox that focused on system call sequences.The authors made a comparison by using static, dynamic and integrated method with using two classifiers which are random forest(RF) and SVM. The accuracy detection shows for integrated method in RF 97.68% while 98.71% using SVM algorithm.

M.S.Anuar & M.Aizaini [12] proposed an improvement decision tree algorithm to classify malware and benign. On binary class they achieved accuracy 94.6% by using API as feature extraction. Liu et al [13], used Neural network for detection. The authors mostly use the static features gained by anti-compilation APK. These static features include string, sensitive API, certificates and application permissions. Yang et al [14], suggested an advanced random forest algorithm to detect and classified malware. Santos et al [15] primarily used the static feature of PE files. In this paper, they recommended a new method to detect unknown malware families based on the frequency of the appearance of opcode sequences.

C. I. Fun et al. [16], suggested techniques of hooking to track dynamic signatures that the malware tries to hide by using data mining methods. This technique detected different behaviors of malware and they compare it with the benign data. By applying 80 attributes, the detection rate was 95% which makes the technique they used increased detection rate with decreasing complexity. M. Belaoued & S. Mazouzi [17] suggested a real-time PE malware detection system based on the analysis of the information stored in the PE-optional header fileds. For features selection, the writers used Phicoefficient and chi square with selected features Rotation forest classifier was trained and tested. Their resulted reached at 97% accuracy. Hassen et al [18] proposed a new technique for malware classification using static analysis based on control statement shingling. In their work, using a dataset of 10,260 malware instances, they reported up to 99.21% accuracy by using disassembled malicious binaries as a extracted features.

In this paper, researchers presents a comparison of malware detection techniques using machine learning algorithm which are K-Nearest Neighbors (K-NN), Decision Tree (DT) and Support Vector Machine (SVM) for malware detection by using portable executable (PE) information as a features extraction. The PE structure contains of a PE file header and a section table followed by the section's data [19]. The results showed that DT is the best machine learning technique to detect malware with 99% detection accuracy. The next sections researchers explain the proposed method and section 3 discusses on the experimental finally are the conclusion and future directions.

2. RESEARCH METHOD

The proposed detection approach is illustrated by the flowchart in Figure 1. A malware sample analyzed using static analysis. Static analysis leads to the extraction of features. In this experiment, we examined PE files with PEview tool. This PE executable files information then will be used as a feature. All the information features will then be filtered to select optimal features that are relevant for classification task. The last process will be done by evaluating the highest accuracy detection using three types of algorithms which are K-Nearest Neighbor (KNN), Decision tree(DT) and Super Vector Machine (SVM).

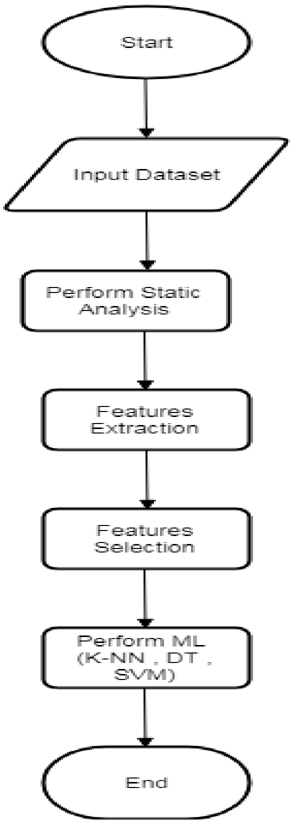


Figure 1. Detection model method

2.1. Data collection

Dataset is divided into malware and benign software. In this study, we collected 305 types of randomly malware and 236 types of benign software. The malware and benign files from malware collectors like VirusShare [20], VirusTotal [21], VX Heaven [22] and etc. For benign files we obtained from Windows and Programs Files folder. All are in the Windows PE format.

2.2. Features extraction

Feature extraction can be defined as transforming the large, vague collection of inputs into the set of features [23]. Advanced detection relies on feature extraction of the malware being analysed [24]. Features could contain plaintext strings found in the disassembled files, the size of the malware, n-gram byte sequences, system resource information such as the set of DLLs, etc. By using machine learning algorithm, these features are provided as inputs. Based on our studies detailed analysis of the format features of the PE files, we extracted about 78 features that have probable to differentiate between clean software and malware, from given PE files. These features are summarized in Table 1.

Table 1. Features to be extracted

Features
Name,e_magic,e_cblp,e_cp,e_crlc,e_cparhdr,e_minalloc,e_maxalloc,e_ss,e_sp,e_csum,e_ip,e_cs,e_lfarlc,e_ovno,e_oemid,e_oeminfo,e_lfanew,Machine,NumberOfSections,TimeDateStamp,PointerToSymbolTable,NumberOfSymbols,SizeOfOptionalHeader,Characteristics,Magic,MajorLinkerVersion,MinorLinkerVersion,SizeOfCode,SizeOfInitializedData,SizeOfUninitializedData,AddressOfEntryPoint,BaseOfCode,ImageBase,SectionAlignment,FileAlignment,MajorOperatingSystemVersion,MinorOperatingSystemVersion,MajorImageVersion,MinorImageVersion,MajorSubsystemVersion,MinorSubsystemVersion,SizeOfHeaders,CheckSum,SizeOfImage,Subsystem,DllCharacteristics,SizeOfStackReserve,SizeOfStackCommit,SizeOfHeapReserve,SizeOfHeapCommit,LoaderFlags,NumberOfRvaAndSizes,Malware,SuspiciousImportFunctions,SuspiciousNameSection,SectionsLength,SectionMinEntropy,SectionMaxEntropy,SectionMinRawsize,SectionMaxRawsize,SectionMinVirtualsize,SectionMaxVirtualsize,SectionMaxPhysical,SectionMinPhysical,SectionMaxVirtual,SectionMinVirtual,SectionMaxPointerData,SectionMinPointerData,SectionMaxChar,SectionMainChar,DirectoryEntryImport,DirectoryEntryImportSize,DirectoryEntryExport,ImageDirectoryEntryExport,ImageDirectoryEntryImport,ImageDirectoryEntryResource,ImageDirectoryEntryException,ImageDirectoryEntrySecurity

2.3. Features selection

Based on Table 1 explained above, there are 78 features were extracted from PE information not all features extracted are significant to be used and will give a high detection accuracy. So, the next phase is we only chose some features from the 78 features. It is done to get more accurate detection. Table 2 shows the most relevant features used and we select only 28 features from 78 features extracted to continue the experiment.

Table 2. Features to be selected

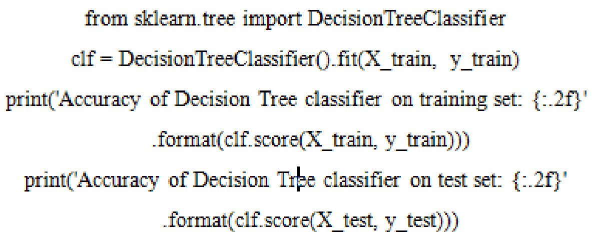
Features
'NumberOfSections','PointerToSymbolTable','NumberOfSymbols','SizeOfOptionalHeader','Characteristics','Magic','MajorLinkerVersion','MinorLinkerVersion','SizeOfCode','SizeOfInitializedData','BaseOfCode','ImageBase','SectionAlignment','FileAlignment','SizeOfHeaders','SizeOfImage','Subsystem','SizeOfStackReserve','SizeOfStackCommit','SizeOfHeapReserve','LoaderFlags','SectionsLength','DirectoryEntryImportSize','DirectoryEntryImport','DirectoryEntryImportSize','DirectoryEntryExport'

2.4. Designing detection model

In this phase, for designing a detection model, machine learning techniques called K-NN, DT, and SVM have been used. The choice of classifier or algorithms depends on the type of features, dataset size and also problem to be solved. These classifiers have been used after removed irrelevant features extraction. Next, these features selection will be trained and tested on each classifier to perform classification task.

3. RESULTS AND DISCUSSION

Figure 2 shows the DT algorithm was used and selected in this experiment which give the best high accuracy detection compared to the others machine learning algorithm in classifying between benign software and malware. Table 3 shows the classification results using K-NN ,DT and SVM algorithms. To evaluate the results, the main performance metrics [25] namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) will be calculated. True Positives Rates (TPR) will give the percentage of correctly identified as malware samples. False Positive Rates (FPR) will give the percentage of wrongly identified as malware samples. The performance metrics are calculated as follows TPR=TP/(TP+FN) and FPR=FP/(FP+TN). While proportion of the total number of predictions that are correct called as overall accuracy that will be computed as Accuracy=((TP+TN))/(TP+FP+TN+FN).



```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier().fit(X_train, y_train)
print('Accuracy of Decision Tree classifier on training set: {:.2f}'
     .format(clf.score(X_train, y_train)))
print('Accuracy of Decision Tree classifier on test set: {:.2f}'
     .format(clf.score(X_test, y_test)))
```

Figure 2. Decision tree algorithm

Table 3. Classification result

Method	TPR	FPR	Detection accuracy (%)
DT	1.00	0.021	99
K-NN	0.92	0.042	94
SVM	0.95	0.160	91

Explaining research chronological, including research design, research procedure (in the form of algorithms, Pseudocode or other), how to test and data acquisition [6-9]. The description of the course of research should be supported references, so the explanation can be accepted scientifically [4, 10].

Tables and Figures are presented center, as shown in Table 1 and Figure 1 and cited in the manuscript before appeared. Based on Table 3 above, we can see that DT algorithm have high detection accuracy and effective to detect the malware by using the proposed dataset for this work compares to the others algorithm which is reached 99%. The performance SVM only 91% is not good enough accuracy. Moreover the number of FPR is also higher 0.160. For K-NN the accuracy percentage detection accuracy is 94%.

4. CONCLUSION

Malware are becoming widespread and more complex day by day. In this experiment, the focus lies on analysing and measuring the detection accuracy of the ML classifier that used static analysis to extract the features based on PE information by comparing three different classifiers on machine learning methods. We were able to train machine-learning algorithms to detect malware and benign files. The results showed that DT machine learning technique is the best classifier to classify our data with 99% of accuracy. From this experiment it is clear that by using static analysis based on PE information and selected the relevant features of the data can also give the best detection accuracy and can accurately represent malware. Furthermore, the advantages of this method there is no need to execute or run malware and we can understand whether it is malware or not.

ACKNOWLEDGEMENTS

The author hereby acknowledges the financial support from University Technology MARA (UiTM) and the Ministry of Higher Education under 600-IRMI/FRGS 5/3 (017/2017).

REFERENCES

[1] Mohammad I , Muhammad Ta, & Muhammad Aq, “A Comparison Of Feature Extraction Techniques For Malware Analysis”, *Turkish Journal Of Electrical Engineering & Computer Science*, available online: https://pdfs.semanticscholar.org/d6ba/bbdd779de6c75abae95a54b4f8dd706d70ef.pdf, last visit:15/10/2018.

[2] G.Bala Krishna, V. Radha, K. Venugopala Rao, “Review of Contemporary Literature on Machine Learning based Malware Analysis and Detection Strategies” *Global Journal of Computer Science and Technology*, [S.l.], july 2016. ISSN 0975-4172. Available at: https://computerresearch.org/index.php/computer/article/view/1410, Date accessed: 23 mar. 2019.

[3] A. . Fallis, “Neural Network Model,” J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2013.

[4] Mohammad DK, Mohd TS, Rafia A, Mahenoor S,& Sonalii S” Malware detection using Machine Learning Algorithms” *IJARCCE*, Vol. 6, Issue 9, September (2017), available online: https://ijarcce.com/upload/2017/september17/IJARCCE%2035.pdf, last visit:20/10/2018.

[5] J.Landage & M.P.Wankhade “Malware And Malware Detection Techniques:A Survey” *International Journal of Engineering Research & Technology (IJERT)*, vol.2 Issue 12, December 2013.

[6] S.Najari & I.Lotfi “Malware detection using data mining techniques” *International Journal of Intelligent Information Systems*, Volume 3, Issue 6-1, December, Pages: 33-37. 2014.

[7] V. Rao & K.Hande “A comparative study of static, dynamic and hybrid analysis techniques for android malware detection” *IJEDR* Vol 5, Issue 2 (2017) available online: https://www.ijedr.org/papers/IJEDR1702223.pdf , Date accessed: 23 March 2019.

[8] Tian R, Islam R, Batten L, Versteeg S. “Differentiating malware from cleanware using behavioural analysis. In: Malicious and Unwanted Software (MALWARE*)”, 2010 5th International Conference* on; New York, NY, USA: IEEE. pp. 23-30. 2010.

[9] Zahra S, Mahboobeh G & Ashkan S “A miner for malware detection based on API function calls and their arguments”, *16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP),* (May), pp. 563-568. 2012.

[10] Arshi D & Maninder S “Behavior analysis of malware using machine learning”, *2015 Eighth International Conference on Contemporary Computing (IC3),* available online: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7337021, last visit: 18/10/2018.

[11] P. V. Shijoa,& A. Salim “Integrated static and dynamic analysis for malware detection” *International Conference on Information and Communication Technologies (ICICT 2014),* pp. 804 – 811, availbe online: www.sciencedirect.com

[12] M.S.Anuar and M.Aizaini “*Classification of Malware Family Using Decision Tree Algorithm”* Innovations in Computing Technology and Applications, vol ,2. 2107.

[13] Liu Yang “Employing The Algorithms Of Random Forest And Neural Networks For The Detection And Analysis Of Malicious Code Of Android Applications”, Beijing Jiaotong University. 2015.

[14] Yang Hong-Yu, Xu Jin.”Android Malware Detection Based On Improved Random Forest”, *Journal On Communications*, 2017,(04):8-16

[15] Santos I, Brezo F, Ugarte-Pedrero X, et al "Opcode sequences as representation of executables for data-mining-based unknown malware detection", *Information Sciences*, 231: 64-82. 2013.
[16] C.-I. Fan, H.-W. Hsiao, C.-H. Chou, and Y.-F. Tseng, "Malware Detection Systems Based on API Log Data Mining," *2015 IEEE 39th Annu. Comput. Softw. Appl. Conf.*, vol. 3, pp. 255–260, 2015.
[17] M.Belaoued & S.Mazouzi "A Real-Time PE-Malware Detection System Based on CHI-Square Test and PE-File Features" 2018.
[18] Hassen et.al, " Malware classification using static analysis based features" In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. 2017.
[19] J.Bai et.al "A malware Detection Scheme Based on Mining Format Information" *The Scientific World Journal* available online: http://dx.doi.org/10.1155/2014/260905, 2014.
[20] https://virusshare.com/
[21] https://www.virustotal.com/
[22] Http://83.133.184.251/Virensimulation.Org/
[23] S.Ranveer & S.Hiray, "Comparative Analysis of Feature Extraction Methods of Malware Detection," *IJCA*, vol.120, June 2015
[24] Qu & Hughes "Detecting metamorphic malware by using behavior-based aggregated signature". Internet Security (WorldCIS), *World Conference Proceedings*, 13–18. 2013.
[25] M.Asha Jerlin & K.Marimuthu "A New Malware Detection System Using Machine Learning Techniques for API Call Sequences" *2017 Journal of Applied Security Research*, vol 13, pp.45-62, 2017.

BIOGRAPHIES OF AUTHORS

Nur Syuhada Selamat obtained her Degree of Netcentric Computing at Universiti Teknologi MARA (UiTM), Malaysia.Now, she is furthering her study at the Faculty of Computer and Mathematical Sciences, in Master of Computer Science at the same university.

Fakariah Hani Mohd Ali obtained her PhD of Security in Computing from Universiti Putra Malaysia. She is a Senior Lecturer at Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia. She is a member of the Malaysian Society Cryptology Research (MSCR). Her research interest are cryptography, network security and digital forensics.