Research Article

Malware Classification Using Machine Learning Algorithm

Ucu Nugraha[1], Azuan Ahmad [2, *], Widyatama[3] ,Wan Nur Aaisyah Wan Mansor [4], Madihah Mohd Saudi [5]

[1]Widyatama University
[2]CyberSecurity and Systems (CSS) Research Unit, Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Negeri Sembilan, Malaysia
[3]Widyatama University
[4]CyberSecurity and Systems (CSS) Research Unit, Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Negeri Sembilan, Malaysia
[5]CyberSecurity and Systems (CSS) Research Unit, Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Negeri Sembilan, Malaysia
[2]azuan@usim.edu.my

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: The rise of malware has resulted in many concerns and trends for future cybercriminals that infect victims' computers to steal information. The majority of the devices are highly vulnerable to simple attacks based on weak passwords, unpatched vulnerabilities, and poorly monitored. Thus, it is the best projection that computers nowadays being the main target to propagate malware. Besides, there is a lack of studies that provide in-depth analysis on malware, especially in the classification model. As there are lots of machine learning algorithms that can be used to detect malware. Besides, a lightweight model is required for the malware detection algorithm to maintain its accuracy without sacrificing the performance. As a solution, we propose a classifier model based on machine learning that can detect malware activities. This research aims to study the existing malware classification algorithm's features, apply the existing algorithm, and evaluate the algorithm for malware classification. This algorithm can detect the malware with high accuracy up to 99.3%. The output of this research will be significantly used in detecting malware attacks that benefit multiple industries including cybersecurity contractors, oil and gas, water, power and energy industries which align with the National Cyber Security Policy (NCSP) which address the risks to the Critical National Information Infrastructure (CNII).

Keywords: malware, algorithm, machine learning, artificial intelligence

1. Introduction

In a world moving rapidly toward the technology-based economics of the 21st century, technologies are rapidly evolving towards a better future due to the rapid development of the 4th Industrial Revolution. At the same time, the increasing number of available malware source code online gives benefits towards the criminal to increase the complexity of malicious code to improve the obfuscation to decrease the chances of being detected by anti-virus programs. This leads to multiple forks or new implementations of the same type of malicious software that can propagate out of control. Based on AV-Test, approximately 390,000 new malware samples are registered every day which gives rise to the problem of processing the huge amount of unstructured data obtained from malware analysis [13]. This increases the challenge for anti-virus vendors to detect unknown malware and release malware signatures in a reasonable time-frame to prevent infection and propagation.

Malware is a type of software that disrupts computer operations without the owner's consent. There are various types of malware such as virus, worm, trojan horse, and ransomware. The virus replicates itself relentlessly and infects files and programs to destroy valuable data or cause irreparable damage, worms is a malicious code that works by propagating itself and distribute through the computer and the infected network, Trojan Horse works by sneaking into victims' computer and act as legitimate program and ransomware works by encrypting important data at the victim's host and ask for ransom from the victim as an exchange for the data.

This study will develop a malware classification algorithm using a machine learning algorithm in classifying malware, based on the given dataset.

2. Literature review

2.1 Malware

Malware is malicious software that can give threat and harmful to the computer user. Malware is developed by attackers to disrupt computer operations and gain unauthorized access to the network. It may appear in terms of code snippets, active content or scripts that directly installed in the device without consent [7]. Malicious actors

distribute malware for unauthorized system access, steal authentication information, sell access to a compromised system, and use it to spam and hold victim file ransom. The increase in the use of cryptocurrency by the public has created another opportunity for cybercriminals to earn money by deploying crypto-mining malware. An attack involving crypto-mining malware is known in the cybersecurity community as a crypto-jacking attack

2.2 Malware Types

Many types of malware are divided into several classes with their purpose. The malware classes are as follow:

- Virus. A small computer program that can replicate itself and infect the computer without user permissions and knowledge. The virus commonly exists in the executable file which spread when the user executes the file [8].
- Worm. Similar to the virus but the worm can spread the malware over the network and replicated it to other machines without user interaction all the time. This kind of issue was generated a lot of network traffic that crushed the internet of the time [9].
- Trojan. A program that pretends to be as legitimate software and acts as a backdoor that persuades the user to execute or install the software. Trojan needs user interaction in cases to spread the malware [9].
- Adware. Malware that acts to display commercial on the computer that will download advertisement automatically based on user information. It works by pretending to be the legitimate program to allow users to install on the computer [13].
- Rootkit. The small program is hard to detect and capability to obfuscate data by running the networks on an infecting system [13].
- Spyware. Spyware is malicious software that works by spying, gathering information of user personal information from the computer like online banking details, website visited, and password account without user knowledge. The spyware can be disguised and its existence is difficult to detect.
- Backdoor. Provide additional entry to the network that allowed a larger attack surface to the attacker. So, the backdoor is not being used by attackers because it does not give any harm to the computer user.
- Keylogger. Act by the record and store all the data that the user has pressed key. The data record includes passwords, bank card numbers and other sensitive data.
- Ransomware. Act by encrypting the data and demand money from victims if they want a decryption key. When the computer has infected by ransomware, the machine will freeze which users cannot open the files unless paid to the attackers.

2.3 Detection Method

Malware detection techniques are divided into two which are signature-based and behavior-based methods. Malware also has two analysis approaches which are static and dynamic malware analysis.

- Static analysis is acted statically which is without execution of the file and the source code of malware only can be view as reading and trying to interpret the behavioral properties of the files. The benefit of static analysis is it can read all the behavioral malware but it rarely uses because is consuming more time. However, it was mostly used for research purposes for example, when developed signature for zero-day malware [5].
- Dynamic Analysis is a monitor on the file when it is executed and the behavioral properties of the file were interpreted. The benefit of dynamic analysis is much faster than static analysis and this file usually runs in the virtual machine, sandbox [5].
- The signature-based analysis is a static method that depends on the predefined signature for example is fingerprint-like MD5 and SHA1 hashes. The file will analyze first by antivirus and the comparison will be conducted between the sample malware. If there are matching signatures between the comparisons, then the file is target suspicious. It is easier to use this analysis because sample malware can always detect just by using the hash value. However, this issue gives a benefit to the attackers to modify the signature and once the signature is changed, the malware cannot be analyzed and detect more by using pure signature-based detection unless the signature was created [5].
- The behavior-based analysis also known as heuristics-based analysis which does not require a signature to detect the malware. The actual malware behavior is observed during execution, looking for signs of malicious behavior. The combination of the action also can increase the files' level of suspicion. In an easy word, it can detect events where behavior is different from normal behavior [5].

2.4 Machine learning

Machine learning is a branch of artificial intelligence that enable machines to perform task skillfully [11]. Machine learning is about designing algorithms that allow the computer to learn without the need for human experts and programmers [14]. Other than that, machine learning algorithms need data to learn to determine the right answer. The effectiveness of this method allowed us to seen the output of the dataset that has been trained. Therefore, the unknown sample that has been trained also can help in this research to determine malware or benign sample. There are three types of machine learning algorithms.

2.4.1 Supervised Learning

The algorithm needs a target which is a dataset to predict the result. The task uses by this algorithm based on a classification problem. Then, the learner required learning until achieved level accuracy on training data. Example of supervised learning is Regression, Decision Tree, Random Forest, KNN and Logistic Regression [1].

2.4.2 Unsupervised Learning

The algorithm doesn't need a target to predict. This algorithm is more focused on clustering populations in different groups. This algorithm aims to make the computer learn more about the data. An example of unsupervised learning is the Apriori algorithm and K-means [1].

2.4.3 Reinforcement Learning

The algorithm that able the machines to train data in provides a specific decision. The machine is exposed which continued training using trial and error. So, this machine will learn from the past and capture the best solution to produce an accurate decision. An example of this reinforcement learning is Markov Decision Process [1].

2.5 Related Work

There are some related works on malware detection using machine learning. "A Multi-Dimensional Machine Learning Approach to Predict Advanced Malware" focused on predicting the advanced malware that similar to Stuxnet by using four different features of the Regression algorithm. The features include Linear and Polynomial Regression and also Random Forest Regression algorithm. The finding of his research found that the Linear and Polynomial Regression is inefficient with four features while Random Forest Regression provides better predictions with more features [3].

To overcome the potential weakness from the previous approach, Wei Zhong had proposed the Multi-Level Deep Learning System (MLDLS) for malware detection. Each deep learning model in the tree structure was built on the subset of data for a particular group of a malware family. Compared to previous deep learning studies for malware detection, it showed that MLDLS was more capable to handle complex malware data distribution. The performance of malware detection using MLDLS was improved as compared to support vector machine decision tree and single deep learning where the accurateness of detection able them to identify the malware more effectively [15].

Other than that, "Machine Learning Aided Android Malware Classification" had researched to detect and analyze malicious Android Apps by using two machine learning aided which are classification and clustering. The permission-based method was demonstrated and was able to classify malware from goodware in 89% of cases while the source code analysis classification of performance was over 95%. The accuracy rate by using SVM was 95.1% and 95.6% by using the ensemble learning method [10].

Besides, the research from Bonan Cuan had presented how they used machine learning techniques to detect malicious behaviors in the PDF files. The SVM classifier firstly was set up and able to detect 99.7% of the malware. Though the classifier was easy to lure with malicious file and they forged to make it clean. For instance, they had implemented a gradient-descent attack to evade the SVM algorithm and almost 100% successful [6].

Last but not least, "Zero-day Malware Detection based on Supervised Learning Algorithms of API Call Signature" had done research that proposed and developed a machine learning framework using eight different classifier algorithms to achieve accuracy and detect the unknown malware. The SVM with normalized polykernel has performed the best among the eight classifiers evaluated in their study [2].

3. Method

3.1 Overview

The methodology of this project is divided into three phases according to research objective. Phase 1 to study the features of existing malware classification algorithm. Phase 2 to apply existing algorithm for malware classification algorithm while in phase 3 the algorithm for malware classification is evaluated.

Table . Summary of Methodology Research

Objective	Research Method	Expected outcome

Phase 1: To study the features of existing malware classification algorithm	1. Define research area, problems, objectives and scope 2. Review the literature on malware and algorithm for malware classification	The summary of classification algorithm for malware classification
Phase 2: To apply existing algorithm for malware classification algorithm	1. Identify the scope of algorithm for malware classification 2. Identify the tool needed to build malware classification algorithm 3. Apply existing algorithm for malware classification	Algorithm for malware classification algorithm is apply to classify malware activity accurately and more affective
Phase 3: To evaluate algorithm for malware classification	1. User testing the algorithm for malware classification	Effective algorithm that can detect malware activity

3.1.1 Phase 1: To study the features of existing malware classification algorithm

In this phase, literature review was reviewed on related works and identified gap. Besides, the objective and the scope of this proposed project also was defined as well. Next, literature review in Chapter 2 also was defining the current problem that related. Other than that, literature review also was discussed about machine learning, method that has been used by attacker to create malware and the algorithm that has been used to detect the malware classification.

The source of the literature review that covers related work on malware and the machine learning algorithm mostly coming from the articles, journals and websites. As a result, the efficient machine learning algorithm for malware classification has been chosen based on the previous study that being reviewed.

3.1.2 Phase 2: To apply existing algorithm for malware classification algorithm

Phase 2 consist of three phases as illustrated in Figure 1 which are data collection, feature selection and extraction and, machine learning classifier. The first stage which is data collection, collect the dataset of malware and benign applications that suitable for machine learning training and testing. The selection and extraction of the features in the second phase are extracted from various behavior and properties of collected malware which is then are labelled and store in Comma Separated Value (CSV) file. The machine learning classifier entails the final phase, whereby the information in the database trains the machine learning classifier to produce a detection model. The following sections communicate each phase in detail.

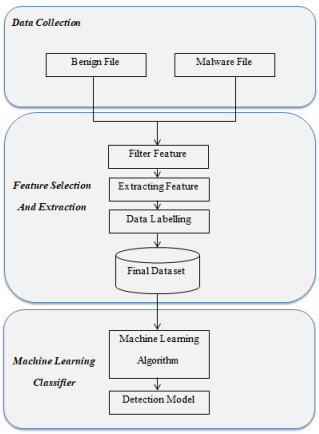

Figure 1. The experiment work flow structure.

3.1.2.1 Data collection phase

In this phase, as explained in Figure 1, the phase start with the collection and identification of benign and malware file. Malware samples can get from several trusted websites which includes VirusTotal and Kaggle.

3.1.2.2 Feature selection and extraction phase

In this phase, we filtered the features of benign and malware files collected during the data collection phase. The features were chosen from numerous network features in the packet-level features. Above all, the main challenge in feature selection is finding the most relevant features that led to the highest true positive rate. A large number of features in the dataset should be filtered and refined. In addition, some features correlate to each other, and this hinders the malware classification process. Moreover, some features may contain redundant information

from other features that will increase computational time and reduce classification accuracy. The extracted features later were stored as a sequence of comma separated values (CSV) files.

3.1.2.3 Machine Learning Classifier Phase

In the final stage, that is the machine learning classifier phase, the classifiers' result is produced. This phase determines the finest machine learning classifier for malware based on performance results. In this step, before determines the best machine learning algorithm firstly, we chosen several machine learning algorithms based on literature review that has been state in Chapter 2. Next, we identify the best parameter setting for each machine learning that has been state so that we can produce accuracy malware detection. Then, the dataset will be loaded into the machine learning algorithm for the training and testing. After the dataset has been loaded, the dataset will be training and testing with each of the selected machine learning algorithm. Lastly, we will monitor the result which is the best algorithm for machine learning classifier will be recorded.

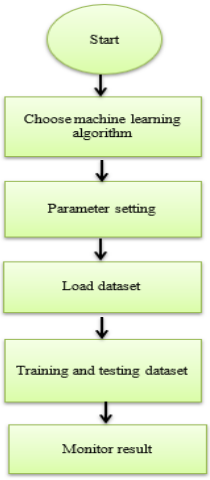

Figure 2. Machine learning classifier process

3.1.3 Phase 3: To evaluate algorithm for malware classification

In this phase it will focus on evaluation of the algorithm for malware classification. In order to evaluate the performance, we used confusion metrics which consist of True Positive, False Positive, True Negative and False Negative. Based on the confusion metric, we derived the False Positive Rate (FPR), False Negative Rate (FNR), True Positive Rate (TPR) and True Negative Rate (TNR).

The expected outcome for this phase is the algorithm that has been proposed can detect the malware and minimize the malware activity. This shows that the algorithm for malware classification algorithm is effective to detect the malware activities.

3.2 Development Tools

In this stage, the application will be created, a few devices have been utilized to plan and coding the framework. The instruments appear in the table underneath facilitate the researcher to build up the framework.

Table 2. Software and hardware development tools

Software / Hardware	Description
Jupyter Notebook	Opensource code to run the coding
JetBrains PyCharm	Integrated Development Environment (IDE) use in computer programming to run a code for Python language
Notepad++	The text and source code editor used in the project
Microsoft Windows 10	Operating System that used in computer to run and test the application
HP Core™ i5-320M @ 2.60GHZ	Computer used to develop the application
MS Word 2010	Used for documentation
MS Excel	Used for key in data
WinZip	Unzipp compressed file

4. Implementation

4.1 Data collection phase

In this phase, the phase started with the collection and identification of benign and malware file. Malware samples coming from several trusted websites which includes VirusTotal and Kaggle [16]. The dataset used for this study consist of 138,047 samples with 41,323 normal samples and 96,724 malware. Figure 3 shows the percentage of benign and malware dataset.

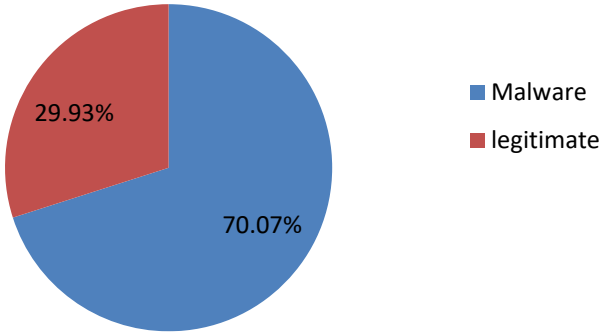

Figure 3. Distribution dataset

4.2 Feature selection phase

The dataset consists of two dimensional 138,047 entries of combination between malicious and benign application and 56 features had been extracted from each of the samples. From our study, 56 features are not feasible for machine learning training and testing thus feature selection should be perform to select best quality feature that fits with our machine learning training and testing.

In obtaining the best features, we used Extra Trees Classifier algorithm for this purpose. Extra Trees Classifier is an ensemble learning method fundamentally based on decision trees and applicable for feature selection. Based on the result, we able to reduce the features from 56 features to 13 best features based on the importance values produced by Extra Trees Classifier algorithm as showed in Table 3.

Table 3. Selected features

Features	Importance Value
DllCharacteristics	0.12405669226638494
Characteristics	0.12290744011093508
SectionsMaxEntropy	0.09944400142308119
VersionInformationSize	0.09366836714692547
MajorSubsystemVersion	0.07301296036459513
Machine	0.06335850808839966
Subsystem	0.06174919561648476
ResourcesMinEntropy	0.04386165355695756
ImageBase	0.04234620214989661
ResourcesMaxEntropy	0.04005470252941394
SizeOfOptionalHeader	0.03908837763852824

MajorOperatingSystemVersion	0.030439149208023 79
ResourcesMinSize	0.021493221879015 85
DllCharacteristics	0.124056692266384 94
Characteristics	0.122907440110935 8
SectionsMaxEntropy	0.099444001423081 19
VersionInformationSize	0.093668367146925 47
MajorSubsystemVersion	0.073012960364595 13
Machine	0.063358508088399 66
Subsystem	0.061749195616484 766
ResourcesMinEntropy	0.043861653556957 56
ImageBase	0.042346202149896 61
ResourcesMaxEntropy	0.040054702529413 94
SizeOfOptionalHeader	0.039088377638528 224
MajorOperatingSystemVersion	0.030439149208023 79
ResourcesMinSize	0.021493221879015 85
DllCharacteristics	0.124056692266384 94
Characteristics	0.122907440110935 8

4.3 Machine learning classifier phase

This phase train and validate the dataset with selected features with two machine learning algorithms namely Random Forest and Gradient Boosting algorithm.

- Accuracy Random Forest = 99.41687794
- Accuracy Gradient Boosting= 98.73234335

From the research that has been done, Random Forest is more accurate than Gradient Boosting which is 99.42 and 98.73. Figure 4 shows the accuracy of both algorithms.

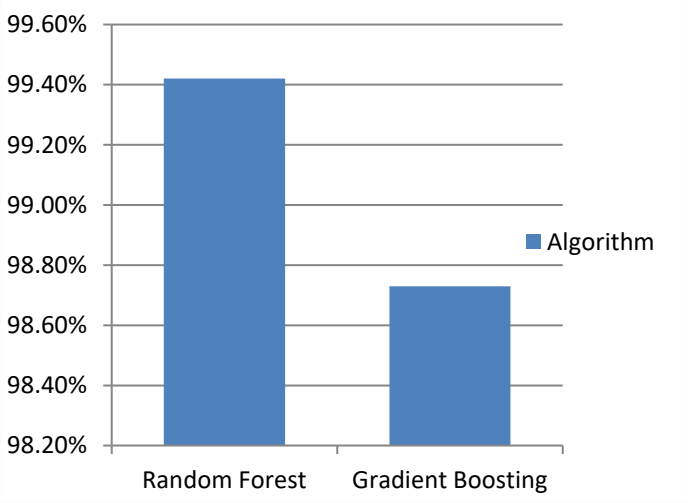

Figure 4. Accuracy of Random Forest and Gradient Boosting algorithm

Besides that, we also can test the confusion matrix to calculate the true positive rate to ensure this algorithm is works accurately. As we known, confusion matrix is well known in the field of machine learning and classification problem that allow visualization of the performance of algorithm.

Table 4 summarizes true positive rate for both algorithm. Random Forest algorithm has highest value true positive rate compare to Gradient Boosting algorithm which is 99.64 and 99.05. True negative rate for Random Forest is 98.89 highest than Gradient Boosting algorithm which is 97.98. False positive rate for Random Forest is less than Gradient Boosting which is 1.11 and 2.02. Lastly, the false negative rate for Random Forest also less which is 0.83 while Gradient Boosting is 2.21.

Table 4. True positive rate for both algorithms

	Random Forest	Gradient Boosting
True Positive	99.6427092	99.05343196
True Negative	98.89129911	97.98236076
False Positive	1.108700892	2.017639241
False Negative	0.833836858	2.206680333

5. System testing and implementation

5.1 Unit testing

Unit testing is a software development process which smallest testable part in application. It can be done during manually or automatically. This system testing is conducted by insert data to login and upload file to check it whether it malicious or not.

5.1.1 Login page

Objective: This testing is to verify the login form is working and check the correctness of output. User need to enter username and password correctly. If users enter the wrong password a username, the error system will be detected.

System testing: If users enter the wrong password, it will be return to the original page again as shown in Figure 5.

Figure 5. Main page of system when user enter the wrong password

5.1.2 Sample upload

Objective: This testing is allowing user to upload their file to check whether it is malicious or not. The sample upload can accept all file that in format csv, exe, jpg, png, doc, bin, pptx and many more. Figure 5.2 shows the interface of sample upload.

1. System testing: User can choose the file that want to be uploaded. The sample upload accept the file in various format. User can upload their file in various formal as the sample upload can accept all fie in format csv, exe, jpg, png, doc, bin, pptx and many more.

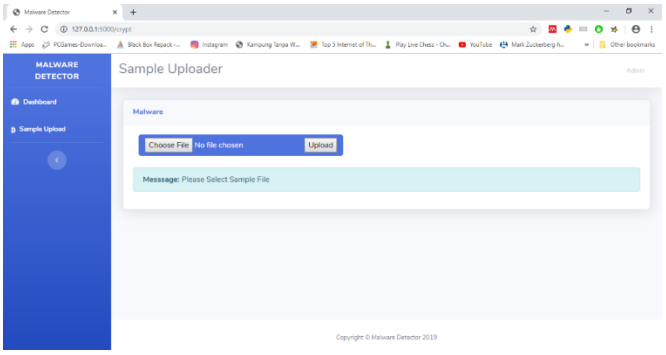

Figure 6. Interface of sample upload

2. System testing: After the user uploads the file, the information and details of the file will be popup. The output message of the file will be appearing too specific whether the file is malicious or not. If the file is malicious, the system will popup message of Message Digest 5 (MD5) hash value, malware hit count, SHA1 value, SHA256 value and the file is malicious. The details of malicious file can be download through the page in CSV format which is contains the details of malware that was hit the file. Figure 7 shows the interface of sample upload with malicious file.

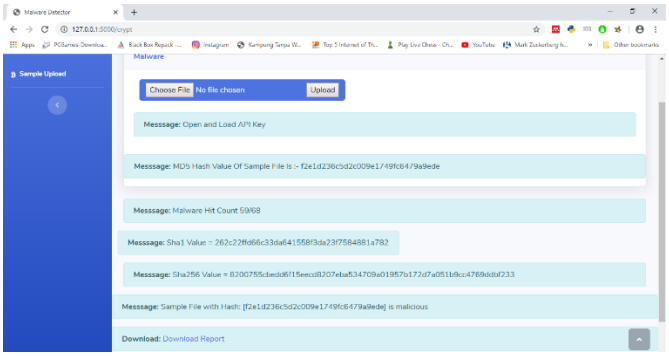

Figure 7. Interface of sample upload with malicious file

Figure 7 shows the interface of sample upload when the user uploads a malicious file. MD5 hash value is message-digest algorithm that commonly use hash function generating a 128-bit hash value. The list of the malware that has been hit can be downloaded in CSV format. SHA1 value is cryptographic hash function that takes an input and generates a 160-bit hash value known as a decrypt message while sha256 value is hash function that generating unique and fixed 256-bit hash value. This file shows that it is malicious as it has been hit by malware that can bring a harmful to the computer.

3. System testing: If the users upload not malicious file, the system will popup message of MD5 hash value and message the file is not malicious. Figure 5.4 shows the interface of sample upload with no malicious file.

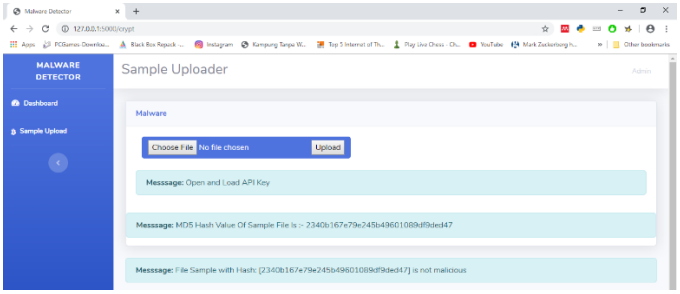

Figure 8. Interface of sample upload with no malicious file

Figure 8 shows the interface of sample upload when the users upload the not malicious file. The outputs for no malicious file will only popup the MD5 value and status of not malicious.

5.2.3 Logout page

Objective: This testing is to verify the logout form is working and check the correctness of output. User need to click logout button and the message 'Select "Logout" below if you are ready to end your current session.' with two choices which are 'Logout' or 'Cancel'. If user choose logout, their access will be terminated but if user choose cancel, user will remain in the system. Figure 9 shows the interface of choices to logout.

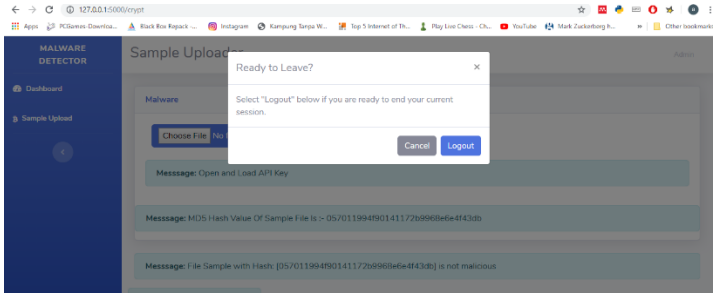

Figure 9. Popup window

6. Conclusion and recommendation

6.1 Objective analysis

The objective of this project was analyzed and reviewed to see whether these objectives have successfully fulfilled or otherwise.

1. To study the features of existing malware classification algorithm

In this project, the literature review that relate to this topic was reviewed. The related work covered on machine learning and algorithm for malware classification. Based on the idea propose by the define author, the features of the existing malware classification algorithm identified and analyze.

2. To apply existing algorithm for malware classification algorithm

Based on the literature review, we have choose two algorithm to further this project which is Random Forest algorithm and Gradient Boosting algorithm. Then, the best parameter setting for both algorithm was identified so that we can produce accuracy malware detection. Next, we upload the dataset into each algorithm for the training and testing. Lastly, we will monitor the result which is the best algorithm for machine learning classifier will be recorded. Random Forest and Gradient Boosting algorithm was applied to classify malware activity accurately and more affective. Both algorithms are monitored and the best algorithms are recorded.

3. To evaluate algorithm for malware classification

We evaluate the performance for Random Forest and Gradient Boosting algorithm by calculating the Accuracy, False Positive Rate (FPR), False Negative Rate (FNR), True Positive Rate (TPR) and True Negative Rate (TNR). We evaluate that Random Forest Algorithm is more accurate that record 99.42% compare to Gradient Boosting that record 98.73%. Therefore, the algorithm can detect the malware effectively and more efficient.

6.2 System strength

This system can detect malware efficiently and accurately to differentiate whether the sample is malicious or not based on calculation of confusion matrix. Next, this system also can save time compare to other system that need to wait for a few minutes longer.

6.4 System limitation

The main limitation of this project is only admin can access the system as there are only one user. Next, the report for sample upload of malware file only can be downloading in format Comma Separated Value (CSV). Then, only file that contain malicious code can be downloaded.

6.5 Future work

In the future, this system can be access by public user so that they can secure their computer from malicious file that can be harm. Furthermore, not only file that contains malicious can be downloaded but not malicious file also can be downloaded too.

6.6 Conclusion

As a conclusion, the main idea to develop malware classification using machine learning algorithm had already been discussed. The idea come based on the problem that had been faced in 4th Industrial Revolution. This system has its own benefit and also limitation. This thesis required determination and a lot of skills before the system and thesis writing are being completed. The result for the accuracy shows that Random Forest algorithm is more accurate than Gradient Boosting algorithm which scores 99.43%. The proposed system is efficient as it can detect malware and benign file accurately and successfully.

Acknowledgment

The authors would like to express their gratitude to Widyatama University, Indonesia and Universiti Sains Islam Malaysia (USIM) for the funding, support, and facilities provided.

References

1. Abdi, A. Three types of Machine Learning Algorithms List of Common Machine Learning Algorithms, 2016. https://doi.org/10.13140/RG.2.2.26209.10088
2. Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. Zero-day malware detection based on supervised learning algorithms of API call signatures. Conferences in Research and Practice in Information Technology Series, 121, 171–182, 2010.
3. Bahtiyar, Ş., Yaman, M. B., & Altınığne, C. Y. A multi-dimensional machine learning approach to predict advanced malware. Computer Networks, 160, 118–129, 2019. https://doi.org/10.1016/j.comnet.2019.06.015
4. Carlin, D., OrKane, P., Sezer, S., & Burgess, J. Detecting Cryptomining Using Dynamic Analysis. In 2018 16th Annual Conference on Privacy, Security and Trust (PST) (pp. 1–6). IEEE, 2018. https://doi.org/10.1109/PST.2018.8514167
5. Chumachenko, K., & Technology, I. Machine Learning Methods For Malware Detection, 2017.
6. Cuan, B., Damien, A., Delaplace, C., & Valois, M. Malware detection in PDF files using machine learning. ICETE 2018 - Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, 2, 412–419, 2018. https://doi.org/10.5220/0006884705780585
7. Imran, M., Arif, T., & Shoab, M. A Statistical and Theoretical Analysis of Cyberthreats and its Impact on Industries, 2018.
8. Khan, I. An introduction to computer viruses: Problems and solutions. Library Hi Tech News, 29(7), 8–12, 2012. https://doi.org/10.1108/07419051211280036
9. Milošević, N. History of malware, 2013. Retrieved from http://arxiv.org/abs/1302.5392
10. Milosevic, N., Dehghantanha, A., & Choo, K. K. R. Machine learning aided Android malware classification. Computers and Electrical Engineering, 61, 266–274, 2017. https://doi.org/10.1016/j.compeleceng.2017.02.013
11. Mohammed, M. M. Z. E., Bashier, E., & Bashier, M. Algorithms and Applications, 2016. https://doi.org/10.1201/9781315371658
12. Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1–21, 2015. https://doi.org/10.1186/s40537-014-0007-7
13. Wang, W., Tian, G., Zhang, T., Jabarullah, N. H., Li, F., Fathollahi-Fard, A. M., ... & Li, Z. (2021). Scheme selection of design for disassembly (DFD) based on sustainability: A novel hybrid of interval 2-tuple linguistic intuitionistic fuzzy numbers and regret theory. Journal of Cleaner Production, 281, 124724.
14. Schapire, R. Machine Learning Algorithms for Classification.
15. Zhong, W., & Gu, F. A multi-level deep learning system for malware detection. Expert Systems with Applications, 133, 151–162, 2019. https://doi.org/10.1016/j.eswa.2019.04.064
16. Kaggle Inc. Kaggle, 2019. Retrieved from https://www.kaggle.com/nsaravana/malware-detection#Malware%20dataset.csv