Analysis of Feature Importance and Interpretation for Malware Classification

Dong-Wook Kim[1], Gun-Yoon Shin[1] and Myung-Mook Han[2, *]

Abstract: This study was conducted to enable prompt classification of malware, which was becoming increasingly sophisticated. To do this, we analyzed the important features of malware and the relative importance of selected features according to a learning model to assess how those important features were identified. Initially, the analysis features were extracted using Cuckoo Sandbox, an open-source malware analysis tool, then the features were divided into five categories using the extracted information. The 804 extracted features were reduced by 70% after selecting only the most suitable ones for malware classification using a learning model-based feature selection method called the recursive feature elimination. Next, these important features were analyzed. The level of contribution from each one was assessed by the Random Forest classifier method. The results showed that System call features were mostly allocated. At the end, it was possible to accurately identify the malware type using only 36 to 76 features for each of the four types of malware with the most analysis samples available. These were the Trojan, Adware, Downloader, and Backdoor malware.

Keywords: Recursive feature elimination, model interpretability, feature importance, malware classification.

1 Introduction

Cyber attacks are becoming increasingly sophisticated and diverse, with many types of malware being discovered in recent cyberattacks. This kind of sophisticated malware has been dubbed as the "intelligent malware". In order to effectively analyze intelligent malware, we must apply various detection and analysis techniques. Machine learning malware classification has exhibited, in particular, the best efficiency in detection, and is considered very valuable by researchers today. Accordingly, this study was conducted for malware classification based on machine learning using minimal information to quickly and effectively respond to new types of malware. For malware analysis, the minimum information should consist of the most accessible information. There are many analytical

[1] Department of Computer Engineering, Gachon University, Sungnam-si, 13120, Korea.

[2] Department of Software, Gachon University, Sungnam-si, 13120, Korea.

[*] Corresponding Author: Myung-Mook Han. Email: mmhan@gachon.ac.kr.

Received: 08 April 2020; Accepted: 28 July 2020.

CMC. doi:10.32604/cmc.2020.010933

tools available to get minimal information, although it is recommended to use popular methods to get quick analytical information. Many research institutions have tried the above method. In 2013, a threat information sharing system (STIX, TAXII) was established to strengthen cybersecurity. This provided a rich database of cyber threat information. Barnum [Barnum (2012)] was established to standardize and share cyber attack information with the goal of collecting that information. In addition, we have seen projects like MAEC (Malware Attribute Enumeration and Characterization), which comes from MITRE, that defines the behavior of malware, artificial artifacts and attack patterns [Kirillov, Beck, Chase et al. (2011)]. Motivated by the increasing number of malicious codes, we conduct this study to classify malicious code as quickly and accurately as possible. Machine learning based malware identification that uses less learning data and only has access to limited features has been singled out as a necessary development [Vabalas, Gowen, Poliakoff et al. (2019)].

A basic requirement for using machine learning in malware analysis is in the technology to extract key information based on the feature engineering of large amounts of malware. Currently, it is possible to perform feature extraction according to malware type using automated analysis tools [Alejandre, Cortés and Anaya (2017)]. Cuckoo Sandbox [Oktavianto and Muhardianto (2013)], an open-source analysis tool that can be widely applied to the selection of the most relevant features, and analyze their importance, was used in this study to extract behavior information. We divided the extracted information into 5 categories: Registry, Network, File System, System Call, and Miscellaneous. Then, Recursive Feature Elimination (RFE), a machine learning feature selection method, was applied to our data. Decision Tree, Random Forest, and Extra Random Forest classification approaches for multi-class classification and interpretability were selected for use with the RFE-based classification model. After obtaining the classification results, the feature importance according to malware type was assessed by interpreting the level of contribution from each feature. This process was done using the Random Forest model as it had shown the best performance. The results of the malware analysis using the Cuckoo Sandbox showed that the features associated with system calls were the most useful.

In this paper, Section 2 introduces the analytical tools and techniques for malware analysis, Section 3 presents the technologies necessary for each step of the process in this study. Section 4 introduces the components of the step-by-step process proposed. Section 5 presents the results of the experiment, and the paper ends with Section 6, which presents our conclusions and future direction.

2 Related work

In this section, representative malware analysis tools and functions are introduced, and feature selection methods from malware analysis studies are summarized. There are several tools for performing dynamic malware analysis using sandbox technology. CWSandbox [Willems, Holz and Freiling (2007)], DRAKVUF [Lengyel, Maresca, Payne et al. (2014)] and Cuckoo Sandbox [Oktavianto and Muhardianto (2013)] are three representative examples.

In this section, Cuckoo Sandbox is introduced while the technical details for using the RFE feature selection method are summarized.

2.1 Malware analysis tool-cuckoo sandbox

Cuckoo Sandbox [Oktavianto and Muhardianto (2013)] is an open-source software for automating the analysis of suspicious files. To do this, the software monitors process actions while files are executed in a specially isolated environment. It then analyzes any custom components. The components of the automated malware analysis using Cuckoo Sandbox and their functions are as follows:

- Cuckoo host: Responsible for guest analysis, management analysis, traffic dumps and creating reports.
- Virtual Network: A virtual environment for analyzing virtual machines in an isolated environment.
- Analysis Guests: An environment where samples are run and the results of analyzing the sample's actions are sent to the Cuckoo host.

Cuckoo Sandbox consists of a central software that runs and analyzes malware samples, as well as a host computer (the Management Software), and several guest computers (Virtual Computers for analysis). Each piece of analysis is performed in an isolated virtual system. The guest computers each serve as an isolated environment for safely running and analyzing the malware, while the host computer runs the core components of the Sandbox that manages the entire analysis process. Cuckoo Sandbox can execute and analyze malware in a virtual environment and can also be used for real-time monitoring. Through Cuckoo Sandbox, files such as Windows exe, DLL, PDF, MS Office, URL, PHP, etc., can be analyzed, while problems associated with a diverse range of issues are continuously reported by the user community. By using Cuckoo Sandbox, the following analysis results can be obtained:

- Traces of Win32 API calls from all processes created by the malware;
- File creation, deletion, and download by malware execution;
- Memory dump of the malware process;
- Tracking of network traffic in PCAP format;
- Windows desktop screen captures during malware execution;
- Complete memory dump of the computer.

2.2 Research on malware feature selection

This section introduces the diverse methods of extracting and selecting the important features of malware. What will be mainly examined are the details of Recursive Feature Elimination.

Usaphapanus et al. [Usaphapanus and Piromsopa (2017)] classified computer viruses based on binary code. After performing a TF-IDF (Term Frequency-Inverse Document Frequency) technique, 8192 features were extracted. Next, the RFE method was applied to effectively reduce the number of features to a more manageable set. It was found that in the case of using the XGBoost and Random Forest approaches, an average classification rate of 93% would be achieved with 20 features, and an average classification rate of 93% was achieved with 500 features.

Ustebay et al. [Ustebay, Turgut and Aydin (2018)] classified the type of attack for the

CICIDS2017 data set based on Random Forest and the Deep Multi-Layer Perceptron (DMLP) classifiers. After learning 80 features from the dataset, the Random Forest classifier was used to find the 10 most influential features. As a result, the score for the ROC curve decreased by about 1% from 0.97 to 0.96. Ustebay deemed that this has real-life application, as it allows quick judgments to be made based only on a few features.

Zeng et al. [Zeng, Chen, Tao et al. (2009)] conducted a study on classifying handwritten numbers using the Least Squares Support Vector Machine (LS-SVM) and the RFE method. The study was carried out in two ways: eliminating the features of each classifier independently (separate) and adding the features of each classifier for elimination (joint). Zeng et al. [Zeng, Chen, Tao et al. (2009)] noted that the RFE method was chosen after confirming its robustness with respect to overfitting in comparison with the wrapper method. In the above study, it was confirmed that the RFE method could be used in various classification situations in addition to the normal security.

In addition, there are complete search, greedy search, and heuristic search methods that can be used for feature selection. We have investigated, in particular and with interest, the meta-heuristic techniques. These algorithms mimic biological and physical behaviors to achieve global optimization. The Particle Swarm Optimization (PSO) algorithm and the Salp Swarm Algorithm (SSA) are representative examples. Combining these two results in the SSAPSO algorithm that has been studied recently [Ibrahim, Ewees, Elaziz et al. (2019)].

In the field of malware research [Kim, Nguyen and Park (2005)] carried out a study to classify DDoS and normality in IDS data by combining genetic algorithms, while Zhang et al. [Zhang, Peña and Robles (2009)] performed a study on a multi-class classification model called the Multi-Label Naïve Bayes (MLNB) model. These studies were designed to first eliminate unnecessary features by principal component analysis and then select features that were important for classification using a genetic algorithm.

3 Malware importance analysis process

The process proposed in this study can be divided into 4 steps, as shown in Fig. 1.

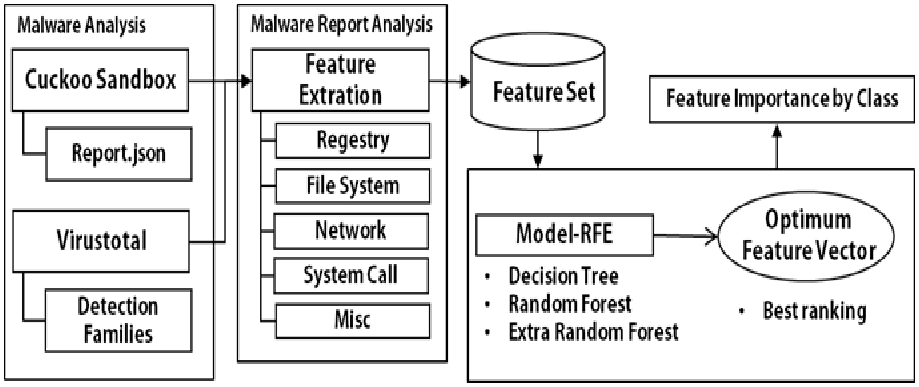

Figure 1: Malware feature importance analysis process

In the first step, malware is analyzed using Cuckoo Sandbox. Cuckoo Sandbox, which is a tool that allows the analysis of malware in a virtual environment can extract various kinds of information (text, network, download, execution, etc.) from the malware and save that information in a JSON format report. Also, in order to determine its exact type, labels for malware type using VirusTotal are collected and categorized. In this study, the feature extraction performed by Korkmaz [Korkmaz (2015)] is referenced, along with the report on the analysis carried out with Cuckoo Sandbox, in the subsequent malware analysis report step, as shown in Tab. 1. Then, feature extraction is performed using the analysis report based on five categories. After generating a dataset based on the extracted features, the features' importance assessed to identify important ones according to the malware type in the third step. In this particular step, when applying the RFE method to identify the important features for each malware, Decision Tree, Random Forest, and Extra-Random Forest classifiers are tested. For the purpose of assessing the importance of each feature according to malware type, a classifier with good analytic ability was selected. In the following step, the feature importance assessment is carried out by selecting only the features with the best ranking, selected by RFE. Best ranking refers to the feature ranked first by the RFE method. In the final fourth step, since it is difficult to evaluate the impact of the best-ranking features ranked first according to malware type, so an impact analysis is carried out for each type based on the Decision Tree, to provide a more detailed importance assessment.

Table 1: Feature extraction category

Category	Feature
Network	Network behavior analysis information
File System	File system manipulation information
Registry	Registry API function information
System call	System call of malware
Miscellaneous	Information on the diverse features of malware

3.1 Recursive feature elimination method

Recursive Feature Elimination (RFE) is one of the most widely used feature selection methods as it reduces the problem's dimensions and boasts high accuracy. It is characterized by repetitive feature selection. There are basically two types of repetitive feature selection: one is to begin without selecting any of the features and then add them one at a time until a certain condition is reached, while the other is to start with all features and eliminate some one by one until a certain end condition is reached. RFE, which falls under the latter category, is a method of creating a model that starts with all the features, then starts eliminating those with the lowest importance, checks the result, then repeats the process to create a new model. The detailed algorithm of RFE is as follows:

1) Train the system on the dataset using all the features;

2) Check the importance of all features;

3) Eliminate the least important feature;

4) Check the accuracy with the set of features after the elimination of one feature;

5) Repeat steps 2 to 4 until the desired condition is satisfied.

As RFE repeats the process of eliminating the feature with the lowest level of importance and checking the result to understand the impact of eliminating that feature. Thus, it is possible to select important features for judgment purposes and avoid feature selection that is simply based on the judgment of importance, in this way it enhances its accuracy. In general, feature selection without prior knowledge can be subjective or ambiguous. Therefore, RFE is frequently used in selecting useful functions [Chen, Meng, Liu et al. (2018)].

3.2 Malware dataset

The PC malware dataset is provided by a manager at the Korea Information Security Agency [KISA (2019)]. It is a "malware profiling system" developed through an information production R&D project at KISA. We used it to evaluate malware variant detection and group classification technology performance. In this study, 5,045 samples were used.

4 Behavior-based malware classification

Behavior analysis is called dynamic analysis, where suspicious files are run and monitored in a controlled environment such as a VM, emulator or simulator [Sihwail, Omar and Ariffin (2018)]. In this section, we describe the extraction of the 5 categories of malware features that relate to Register, Network, File System, System call, and Miscellaneous based on the Cuckoo Sandbox tool.

4.1 Experimental environment

Cuckoo Sandbox [Oktavianto and Muhardianto (2013)] is open-source software for automating the analysis of suspicious files. We used a custom component for monitoring process actions while files are executed in an isolated environment. The malware analysis environment was set up in the way shown in Tab. 2.

Table 2: Experimental environment

Component	Specification
Server	OS: Xenserver7, CPU: Intel Xeon E5-2620 v4, Memory: 48 GB
Cuckoo-Host	OS: Ubuntu16.04 Desktop, Memory: 8 GB
Guest PC	OS: Window8-32bit, Memory: 4 GB

4.2 Malware types and collection

To determine the type of each piece of malware, labels for the malware samples were defined using the VirusTotal API. The VirusTotal API allows users to scan malware files without using a website interface. Access to the VirusTotal API was obtained by using a private API key provided by the VirusTotal Community. The VirusTotal API was used to conduct the experiment based on Kaspersky Lab's labels, these labels are the industry standard. The malware was classified and collected as shown in Tab. 3. It should be noted that the Kaspersky Classification System is used as a classification standard for many anti-virus solutions.

Table 3: Malware collection count

Malware type	Malware Collection Count
Trojan	1960
Adware	1227
Undetected	401
Downloader	375
Backdoor	265
Virus	264
Worm	234
Packed	101
Webtoolbar	50
Risktool	43
DangerousObject	42
Email-Worm	33
Exploit	23
Nettool	17
Net-Worm	10
Total	5045

4.3 Malware feature extraction

In the section, feature vectors were extracted based on the artifacts in the Cuckoo Sandbox analysis report. We analyzed the information about the targets of attack and attack behavior to obtain behavioral information from the Malware. The Malware uses various channels to attack. We can define the malware by dividing it into categories according to its channel of attack, such as through Removable data storage, Web, Local network, etc. [Choi, Lee and Kwak (2019)]. However, in many categories, it was found that there is a problem in terms of the diversity of features for the Malware, so the categories have been redefined using simple features. As shown in Tab. 1, feature extraction was conducted in a total of five categories: network information, registry information, system call information, file system information, and miscellaneous information. The paper published by Korkmaz [Korkmaz (2015)] was referred to when deciding these categories.

First, in the case of network information, a total of 19 network features including UDP, IRC, HTTP, SMTP, TCP, ICMP, DNS connection, and the host and domain name check were extracted from the network-related artifacts. Using these protocols malware can maliciously generate large amounts of traffic [Xiong, Yang, Zhao et al. (2017)]. We extracted this information in real-time to measure the balance of network traffic.

As for the file system, several types of injection and hooking technology were employed while the Cuckoo Sandbox analysis tool was running the malware and has been extracted as a total of 174 features. These features are system artifacts that were detected using file system manipulation in relation to access, file reading, file creation, file modification, and file deletion.

In the case of registry information, Cuckoo Sandbox extracted features by organizing API functions that detected registry operations. These operations were categorized into 4 behaviors: registry key access, reading, modification and deletion.

In the case of System calls, the features can be classified into several types according to the types of operations performed in the Cuckoo Sandbox. These include system and API call tracking, screen capture and browser encryption processes. A total of 312 System call features were extracted that relate to the API file names where API calls begin.

As for Miscellaneous, a total of 403 other functions was extracted from various types of artifacts related to other types of malware include. These include new mutexes, service startup, and execute commands.

5 Experiment results

In this section, we look at the experiments conducted based on the dataset from which five categories were extracted using Cuckoo Sandbox. First, we measured the performance of the tree classifier by selecting the features with the RFE method. The tree classifier was selected as it was found to be the most efficient for interpretation and multiclass classification. Then, based on the tree classifier's results, the features with the highest contribution were analyzed.

5.1 Results of RFE-based malware classification

The number of unnecessary features was reduced for the purpose of assessing the important features for each malware type. To this end, the RFE method, which involves repeatedly eliminating the feature with the lowest importance, was applied to the classification process as the model-based feature selection method. The total number of features and the number of category features prior to using the RFE-based classifier is shown in Tab. 4. In addition, the accuracy of the malware classification results was assessed by applying RFE based on the Decision Tree, Random Forest, and Extra Random Forest algorithms. This served as the basis for determining whether the five category features that were extracted, have been properly classified by the classifier.

In Tab. 5, the number of features selected from the dataset consisting of 804 features in five categories was 291 for Decision Tree, 456 for Extra Random Forest, and 202 for Random Forest. As for the number of features in each category, it was found that all the features of the File System and Network were selected using the Decision Tree and the least used category was classified as Miscellaneous. In the case of Extra Random Forest, System call accounted for a large portion of features, followed by Registry, File System, Network and Miscellaneous. As for Random Forest, System call was the category with the largest number of features, while the Registry accounted for the largest size, followed by System Call, Network, File System and Miscellaneous. What is shown in Tab. 4 is that Random Forest was found to have the best classification accuracy, based on a comparison of using the features selected based on RFE, as shown in Tab. 5. In addition, using Random Forest based RFE, similar performance levels can be identified even if the feature is eliminating about twice as much.

Table 4: Overall feature classification result

	Decision Tree	Extra Random Forest	Random Forest
System call	312	312	312
Miscellaneous	403	403	403
Network	21	21	21
File System	64	64	64
Registry	4	4	4
Full Feature	804	804	804
Running Time(s)	4.997	20.654	15.326
Accuracy	71.94%	76.41%	77.5%

Table 5: RFE based feature selection classification result

	Decision Tree	Extra Random Forest	Random Forest
System call	103	274	144
Miscellaneous	99	130	38
Network	21	7	5
File System	64	41	12
Registry	4	4	3
RFE Selction	291	456	202
Accuracy	71.91%	77.30%	77.9%

The classification results for the importance of the features ranked first were checked by applying the RFE algorithm to the selected features. These consist of features which are ranked first, based on the best ranking, derived by each classifier and only the features whose scores indicate the biggest impact among the 804 features, were extracted. This strategy may be an extreme choice, as it only considers features, judged on the basis of order of priority.

5.2 Interpretation of important features by malware classification

To identify the important functions of malware, we need to interpret the classification results. How we interpret the classification results differs depending on the complexity of the learning model. This is because the ability of the model to fit data for each learning model is measured through predictive accuracy [Murdoch, Singh, Kumbier et al. (2019)]. So, we chose a tree-type classifier with high transparency to select the learning model. In order to select important features according that were to malware classification, unnecessary features were first eliminated. Then, the features judged to be important for each malware type based on the classifier's interpretation were checked.

Table 6: Analysis of important features based on Random Forest

Malware type	Feature importance	System call	Miscellaneous	Network	File	Registry
Adware	76	59	13	2	2	0
Virus	78	55	14	3	5	1
Worm	41	28	10	1	1	1
Downloader	69	48	15	2	3	1
Trojan	68	43	15	2	7	1
Net-Worm	1	1	0	0	0	0
Packed	32	23	4	1	4	0
NetTool	15	11	2	0	0	2
Email-Worm	24	17	6	0	1	0
Backdoor	36	28	6	0	2	0
RiskTool	74	54	14	2	3	1
WebToolbar	92	68	19	0	2	3
Undetected	55	39	11	1	2	2
Dangerous Object	11	7	2	0	2	0
Exploit	48	33	10	2	2	1

Tab. 6 shows the details of the Random Forest classifier that exhibited the best performance among the three classifiers. First, the number of features that affect malware classification was checked. The 15 types of malware with the largest number of the most important features were found to be WebToolbar, Virus, Adware, RiskTool, Downloader and so on. However, a small number of samples shown in Tab. 3 such as WebToolbar were excluded because they were judged to have an invalid number of features and could cause limitations in the generalization of the results. Therefore, Trojan, Adware, Downloader, and Backdoor, for which there were many samples, were examined.

For the majority of the malware types, the features under the category of System call were found to have the biggest influence, followed by the features under the category of Miscellaneous.

Next, these four types of malware were compared by a performance evaluation of all the features and the selected features. When comparing the performance results shown in Tabs. 7 and 8, the difference in the level of selection precision the largest for Backdoor at 7%, but it can be seen that the difference was 3% on average. In the case of the F1-score, the difference was also approximately 3% for Trojan and it exhibited the biggest difference when compared with Backdoor in terms of sample size.

This difference may be due to the difference in the performance evaluation of the recall. However, when compared with 202 features, no significant difference was found in feature selection according to their importance. In addition, classification time could be reduced by 30%-50% over 3-6 seconds depending on the number of features. As a result, it was confirmed that malware classification can be performed even with a small number of features.

Table 7: Performance evaluation of all features of malware

	TROJAN	ADWARE	DOWNLOADER	BACKDOOR
Overall Features	202	202	202	202
Precision	77%	87%	80%	81%
Recall	85%	75%	79%	66%
Running Time(s)	11.564	11.564	11.564	11.564
F1-score	81%	81%	79%	73%

Table 8: Performance evaluation of malware feature importance selection

	TROJAN	ADWARE	DOWNLOADER	BACKDOOR
Feature importance Selection	68	76	69	36
Precision	76%	86%	77%	74%
Recall	80%	75%	80%	67%
Running Time(s)	7.451	8.409	7.456	5.994
F1-score	78%	80%	78%	70%

Fig. 2 shows in detail which features are important for each type of malware. This can be seen through feature contribution and by analyzing the effect of model prediction on individual instances according to the Random Forest [Palczewska, Palczewski, Robinson et al. (2013)]. Therefore, it is possible to understand which features are important for which type of malware. Based on the identification of such features, only the meaningful features, without any unnecessary ones, can be extracted. By analyzing the level of contribution of each feature using Random Forest, only the important features of each type of malware were used in order to speed up the process of malware classification. Also, by ensuring the transparency of the classifier itself, it was possible to identify meaningful features through an assessment of the level of contribution of each feature. If you look at the analysis of the functional contributions in Fig. 2 you can see that the system call in Tab. 6 lead to the selection of most features. However, it was confirmed that the common features among the four types malware in Tab. 8 are MSVCRT.dll and user32.dll that belong to the Miscellaneous feature category. MSVCRT.dll is a basic API that is declared to use a mouse, a keyboard, and a monitor. And user32.dll is a standard for creating and handling the standard elements of the Windows user interface, such as windows and menus of the Windows components. These two are used as a set of basic Win32 API implementations. Next, it is confirmed that FindResourceExW, NtOpenMutant, VERSION.dll belonging to System call and Miscellaneous, also make a high contribution. By analyzing a higher contribution of features as described above, it was found that we could quickly classify malware through identifying the important features of each malware type.

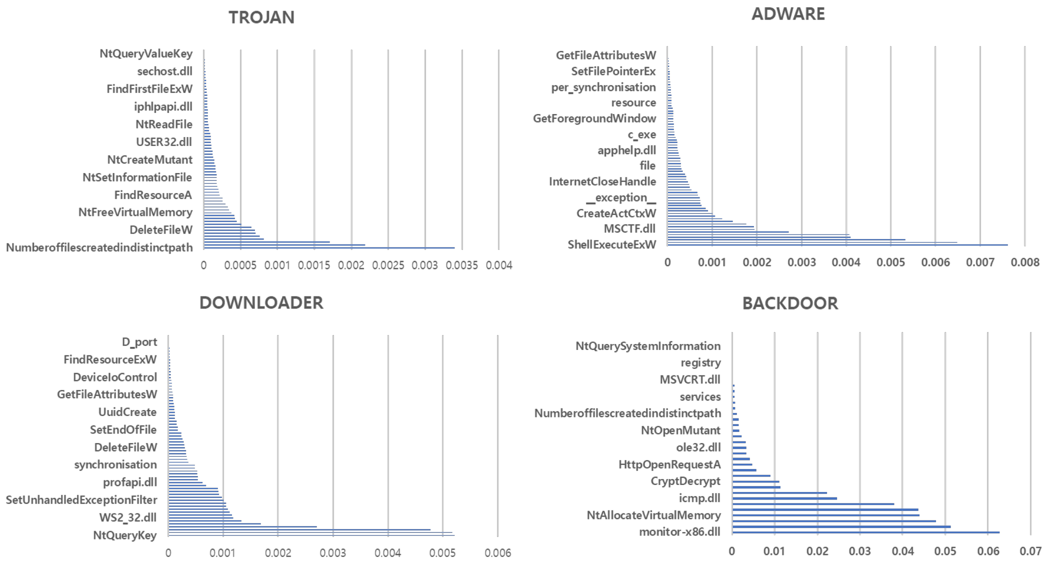

Figure 2: Malware classification feature contribution evaluation

6 Conclusion

For the rapid classification of intelligent malware, this study analyzed the important features of each type of malware. To this end, the 804 features extracted by malware type using Cuckoo Sandbox were split into five categories. But since it would be costly and time-consuming to use such a large number of features, we propose a method to effectively classify malware based on a much smaller number of features. Using the recursive feature elimination method, the number of features examined was reduced while maintaining an accuracy of up to 78% with Random Forest and a margin error of just 0.4%, as shown in Tab. 4. All three classifiers were capable of performing malware classification using all the features at an accuracy of around 70%. An identical performance level was achieved using around 30% of the number of features, after selection based on RFE. In addition, a comparison of the performance assessments with respect to the major types of malware, as shown in Tab. 6, showed that, the marginal error, was around 1 to 3%. The biggest factor for malware came from the category, System call, and classifying the malware type features of the System call category play the most important role in the malware operation.

In the future, we will investigate a feature selection method that is based on GA, not using the RFE method, as mentioned in Section 2. There are also plans to conduct a study on obtaining valid groups based on the correlation between features. RFE is a method for repeatedly selecting the best features, and it was found that such an extreme feature selection method is characterized by functional discrepancies of the different features selected by each classifier during malware classification. There are plans to overcome these current limitations and improve the classification performance by exploring the research data provided by Yang et al. [Yang and Ong (2012)], in order to study the measures for classifiers in order to extract identical features for the same malware type.

Funding Statement: This work was supported by the Research Program through the National Research Foundation of Korea, NRF-2018R1D1A1B07050864.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study

References

Alejandre, F. V.; Cortés, N. C.; Anaya, E. A. (2017): Feature selection to detect botnets using machine learning algorithms. *International Conference on Electronics, Communications and Computers*, pp. 1-7.

Barnum, S. (2012): Standardizing cyber threat intelligence information with the structured threat information expression (STIX). *Mitre Corporation*, vol. 1, no. 1, pp. 1-22.

Chen, Q.; Meng, Z.; Liu, X.; Jin, Q.; Su, R. (2018): Decision variants for the automatic determination of optimal Feature Subset in RF-RFE. *Genes*, vol. 9, no. 6, pp. 1-13.

Choi, S. K.; Lee, T.; Kwak, J. (2019): A study on analysis of malicious code behavior information for predicting security threats in new environments. *KSII Transactions on Internet and Information Systems*, vol. 13, no. 3, pp. 1611-1625.

Ibrahim, R. A.; Ewees, A. A.; Elaziz, M. A.; Lu, S. (2019): Improved SALP swarm algorithm based on particle swarm optimization for feature selection. *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3155-3169.

Kim, D. S.; Nguyen, H. N.; Park, J. S. (2005): Genetic algorithm to improve SVM based network intrusion detection system. *19th International Conference on Advanced Information Networking and Applications*, vol. 2, pp. 155-158.

Kirillov, I.; Beck, D.; Chase, P.; Martin, R. (2011): Malware attribute enumeration and characterization. *Mitre Corporation*, pp. 1-22.

KISA. (2019): *Manager at Korea Information Security Agency.* https://www.kisis.or.kr/kisis/index.do.

Korkmaz, Y. (2015): Automated detection and classification of malware used in targeted attacks via machine learning. Diss. *Bilkent University*, pp. 1-53.

Lengyel, T. K.; Maresca, S.; Payne, B. D.; Webster, G. D.; Vogl, S. et al. (2014): Scalability, fidelity and stealth in the DRAKVUF dynamic malware analysis system. *Proceedings of the 30th Annual Computer Security Applications Conference*, pp. 386-395.

Murdoch, W. J.; Singh, C.; Kumbier, K.; Abbasi-Asl, R.; Yu, B. (2019): Interpretable machine learning: definitions, methods, and applications. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 44, pp. 22071-22080.

Oktavianto, D.; Muhardianto, I. (2013): *Cuckoo Malware Analysis*. Packt Publishing Ltd., pp. 1-142.

Palczewska, A.; Palczewski, J.; Robinson, R. M.; Neagu, D. (2013): Interpreting random forest models using a feature contribution method. *IEEE 14th International Conference on Information Reuse & Integration*, pp. 112-119.

Sihwail, R.; Omar, K.; Ariffin, K. A. Z. (2018): A survey on malware analysis techniques: static, dynamic, hybrid and memory analysis. *International Journal on Advanced Science, Engineering and Information Technology.* vol. 8, no. 4-2, pp. 1662-1671.

Usaphapanus, P.; Piromsopa, K. (2017): Classification of computer viruses from binary code using ensemble classifier and recursive feature elimination. *Twelfth International Conference on Digital Information Management*, pp. 27-31.

Ustebay, S.; Turgut, Z.; Aydin, M. A. (2018): Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. *International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism*, pp. 71-76.

Vabalas, A.; Gowen, E.; Poliakoff, E.; Casson, A. J. (2019): Machine learning algorithm validation with a limited sample size. *PLoS One*, vol. 14, no. 11, pp. 1-20.

Willems, C.; Holz, T.; Freiling, F. (2007): Toward automated dynamic malware analysis using cwsandbox. *IEEE Security and Privacy*, vol. 5, no. 2, pp. 32-39.

Xiong, B.; Yang, K.; Zhao, J.; Li, K. (2017): Robust dynamic network traffic partitioning against malicious attacks. *Journal of Network and Computer Applications*, vol. 87, pp. 20-31.

Yang, J. B.; Ong, C. J. (2012): An effective feature selection method via mutual information estimation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 6, pp. 1550-1559.

Zeng, X.; Chen, Y. W.; Tao, C.; van Alphen, D. (2009): Feature selection using recursive feature elimination for handwritten digit recognition. *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1205-1208.

Zhang, M. L.; Peña, J. M.; Robles, V. (2009): Feature selection for multi-label naive bayes classification. *Information Sciences*, vol. 179, no. 19, pp. 3218-3229.