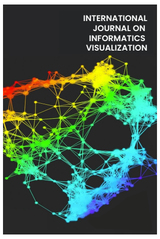
INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



Malware Authorship Attribution Model using Runtime Modules based on Automated Analysis

Sangwoo Lee [a], Jungwon Cho [b,*]

[a] Department of Convergence Information Security, Graduate School, Jeju National University, 102 Jejudaehak-ro, Jeju, Republic of Korea
[b] Department of Computer Education, Jeju National University, 102 Jejudaehak-ro, Jeju, Republic of Korea
Corresponding author: *jwcho@jejunu.ac.kr

Abstract— **Malware authorship attribution is a research field that identifies the author of malware by extracting and analyzing features that relate the authors from the source code or binary code of malware. Currently, it is being used as one of the detection techniques based on malware forensics or identifying patterns of continuous attacks such as APT attacks. The analysis methods to identify the author are as follows. One is a source code-based analysis method that extracts features from the source code, and the other is a binary-based analysis method that extracts features from the binary. However, to handle the modularization and the increasing amount of malicious code with these methods, both time and manpower are insufficient to figure out the characteristics of the malware. Therefore, we propose the model for malware authorship attribution by rapidly extracting and analyzing features using automated analysis. Automated analysis uses a tool and can be analyzed through a file of malware and the specific hash values without experts. Furthermore, it is the fastest to figure out among other malware analysis methods. We have experimented by applying various machine learning classification algorithms to six malware author groups, and Runtime Modules and Kernel32.dll API extracted from the automated analysis were selected as features for author identification. The result shows more high accuracy than the previous studies. By using the automated analysis, it extracts features of malware faster than source code and binary-based analysis methods.**

Keywords— **Malware authorship attribution; automated analysis; runtime modules; machine learning classification.**

Manuscript received 17 Nov. 2021; revised 14 Jan. 2022; accepted 23 Mar. 2022. Date of publication 31 May 2022.

I. Introduction

Recently, with the development of IT technology, there have been positive changes and negative changes. According to the KISA report released in 2021, it is said that various mutant malware is being generated in large quantities, and according to the Verizon report released in 2020, it has found a substitute for the use of single malware like the Trojan horse since 2016 and increased attacks by the Advanced Persistent Threat (APT), which has been attacked for a long time. Therefore, the existing security control system alone has limitations in defense and response to cyber-attacks.

In order to solve this problem, research on malware using artificial intelligence or the attribution of authorship of malware is being conducted. Attribute malware through CNN algorithms is proposed by Kamundala and Kim [1], and [2] studied how malware were detected using R-CNN based on deep learning. In addition, Hong et al. [3] conducted research on the selection of features to classify the group of the authors of the malware, and Shin et al. [4] proposed a framework based on a genetic algorithm that extracts characteristics to attribute the attacker. The study of the attribution of malware authorship is a branch of research extended from existing authorship attribution studies [5]. The reason why the authorship of the malware is attributed is that the profile of the authorship of the malware is developed and the characteristics that only the authorship has, and the obfuscation method is identified so that important information can be provided at the forensic stage of the malware, and the pattern of malware can be identified and the damage can be reduced [6]. Methods of attribution to identify authorship of existing malware include a method of attribution based on source code, from which features are extracted from source code, and a method of attribution based on binary code, from which features are extracted when malware code files are converted into binary [6].

However, the method of attributing existing malware has recently been mass-produced due to modularization and standardization, and there is a limit to time and manpower to apply to the whole of mutated malware. Therefore, in this study, runtime modules, from which these limitations can be extracted by automated analysis to be solved, are used as a feature of attribution for the authorship of the malware.

II. Material and Method

A. Malware Authorship Attribution

Existing authorship attribution studies began in the 19th century with the study of The Plays of Shakespeare by Mendenhall. The study found that the frequency of words frequently used in Shakespeare's plays was statistically analyzed, the characteristics of Shakespeare were attributed, and the author was assigned to unknown works to see if the authorship was Shakespeare [6]. The attribution of malware authorship is that this method has been applied to malware. In the early study of the attribution of malware authorship, the authorship was identified by using features such as the name of the function and the name of the variable in the malware's source code. With the use of these features, the accuracy of the authorship attribution has increased, but the source code data of malware has become difficult to obtain, and even if the file is transferred through a decompiler and converted into source code, it is not possible to change the names of variables or functions that can be selected as characteristics due to the characteristics of the decompiler [7].

The underlying attribution of binary code has been studied to address these limitations. As a method of attribution based on binary code, data can be extracted from malware files in a way that malware files are converted into binary, and the characteristics of attribution are found among them. Features of identification were selected as opcode, binary string, etc. However, due to the nature of binary files, the greater the number of authorships, the more overlapping characteristics, and the greater the time required to extract characteristics. So, there is a limit applied to malware produced in large quantities due to recent modularization and standardization [7].

B. M malware Analysis Method

The malware analysis can be divided into four types according to the method. First, automation analysis is an analysis method used by the tool to analyze malware produced in large quantities due to modularization and standardization. The advantage is that malware can be analyzed quickly, and even if files are missing, with some tools, analysis can be carried out only with hash values. However, due to the automated tool, the accuracy of attribution of malware is reduced compared to other analysis methods, and the file is not directly executed and analyzed, making it difficult for malware with obfuscation technology to be identified. Tools of the analysis include Buster Sandbox Analysis (BSA), Malwares.com, and VirusTotal, and information that can be extracted include communication address, Portable Executable (PE) structure, and Runtime Modules.

Moreover, there is static analysis. Static analysis is a method of analysis in which malware is not executed, and appearance is viewed and analyzed. The advantage is that malware is not executed, making it secure, and more information can be extracted compared to automated analysis. However, it is difficult to identify malware applied with obfuscation technology like automation analysis. Information that can be extracted includes string information in binary and resource information [8].

Next, there is dynamic analysis. Dynamic analysis is a method of monitoring changes as malware is actually executed in virtual environments. The advantage is that malware is analyzed as it is actually executed, so more information can be extracted, and it can also be identified in the case of malware that has been obfuscated. However, this takes a long time to analyze and requires specialized skills as malware is executed and analyzed one by one. Information that can be extracted includes Registry, File system, and Process [8].

Finally, there is a detailed analysis. Detailed analysis is a method of analysis used if there is any shortage as malware is analyzed through existing analysis methods. This has the advantage of extracting information from most malware, but it has the disadvantage of requiring specialized skills. In this study, automated analysis is used to identify the authorship of malware produced in large quantities due to standardization and modularization.

C. Machine Learning Classification Algorithm

This is a kind of AI supervised learning, an algorithm where renewed data labels are determined through existing data and labels. The types of classification algorithms used in this paper include k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), Decision Tree, Naive Bayes, Adaptive Boosting, and Gradient Boosting.

The k-NN algorithm is an algorithm that is determined by finding the nearest k number of data from the new data when the new data is entered into the existing data. This is a distance-based classification algorithm, so it has high accuracy in numerical data, but the more properties to be compared, the slower the classification [9].

The SVM algorithm is an algorithm that is based on existing data and categorized according to the location of the data when renewed data is entered after the area is divided. This is an algorithm in which data with diverse characteristics are classified, and the performance is improved even with less data [10].

The Decision Tree algorithm is an algorithm in which a Tree is generated based on the characteristics of the data, and Tree is passed through and classified when renewed data is entered. This has the advantage of high accuracy, but it also has the disadvantage of being easier to be overfitting [11].

The Naive Bayes algorithm is an algorithm in which the data is assumed to be an independent event and then placed into the Bayes theory to classify. This shows good performance in classifying documents, but it has a disadvantage of low accuracy other than the classification of documents.

The Adaptive Boosting algorithm is an algorithm in which multiple classifiers are created through classifiers of the same algorithm, and the value of the prediction is obtained through weighted voting. This is easy to implement, but it has the disadvantage of being slower than other boosting algorithms [12]. The Gradient Boosting algorithm is an algorithm that learns to a classifier with renewed residual errors through gradient descent, and this tends to perform well among machine learning classification algorithms [12].

D. *Malware Authorship Attribution*

The source code is converted into binary, and idioms, graphlets, and super graphlets are used as features of attribution to cluster and classify the authorship [13]. As shown in Fig. 1, 25 authorships showed 77% accuracy, while the top 5 authorship showed 94% accuracy.

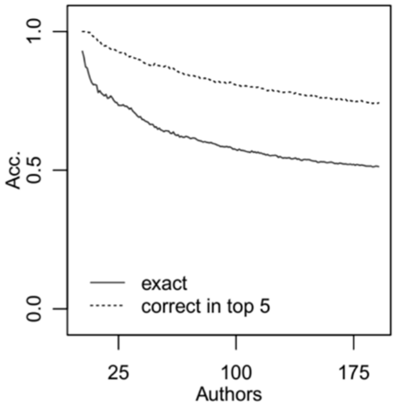


Fig. 1 Experimental Result in Rosenblum et al[13]

However, the source code of the general authorship used in a competition that is not the authorship of the malware is used, which leads to clustering, and there is a limit that clustering can proceed only when the source code data is acquired. Alrabaee et al [14]'s study was supplemented by the existing APPB (Identify the Author of Program Binaries), suggesting three layers of OBA2 (An Onion approach to Binary code Authorship Attribution) methodologies. As shown in Fig. 2, a study was conducted that lowered the inaccurate prediction rate.

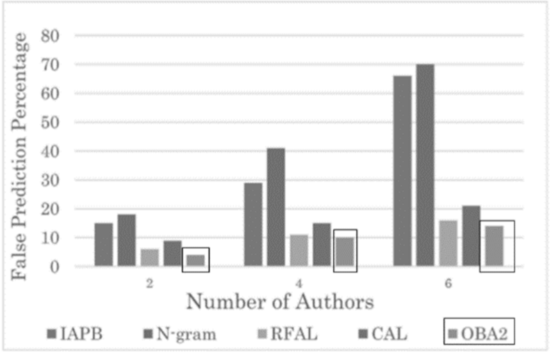


Fig. 2 Experimental Result in [14]

The results of Alrabaee et al. [14] showed higher accuracy than the attribution of existing authorship. However, because there are more features being extracted, there is a limitation that it is difficult to apply to mass-produced malware due to recent modularization and standardization. For Hong et al. [15], 1,944 malware samples were collected, and deep learning research was conducted through a total of 11,144 data sets using source code-based and binary-based identification methods and compared with existing SVM models. The result is shown in Table 1.

The results of Hong et al. [15] are more accurate, but there is a limit that it is difficult to apply to recently mass-produced malware because of the increasing characteristics that need to be extracted. A framework for extracting characteristics of a group of malware attackers based on genetic algorithms is proposed in Shin et al. [4]. This was constructed using a method of source code-based and binary-based attribution, and the results of the experiment are shown in Table 2[4].

TABLE I
EXPERIMENTAL RESULT IN HONG ET AL [15]

	Deep Learning	SVM
Accuracy	94.96%	93.42%
Precision	94.88%	93.32%
Recall	94.96%	93.42%
F-Measure	94.82%	93.12%

TABLE II
EXPERIMENTAL RESULT IN SHIN ET AL. [4]

System	**Number of authors**	**Accuracy**
System in Alrabaee et al [14]	5+	84%
System in Rosenblum et al [13]	50+	78%
Proposed System in Shin et al [4]	5	86%

Research by Shin et al. [4] has increased accuracy depending on the number of authorships compared to previous studies, but it takes time and specialized skills to extract characteristics, making it difficult to apply to recent mass-produced malicious codes.

E. *Malware Authorship Attribution Using Runtime Modules*

This paper suggests that the limitations of the data acquisition of the source code-based analysis used to identify the authorship of the existing malware and the limitations of the extraction of the characteristics of the binary-based analysis are complemented and the authorship can be identified in a short period of time. Fig. 3 is the process of model creation.

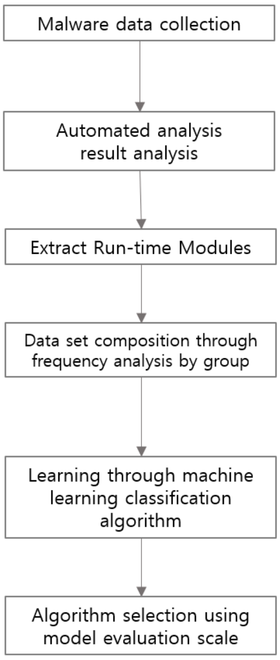

Malware data collection
Automated analysis result analysis
Extract Run-time Modules
Data set composition through frequency analysis by group
Learning through machine learning classification algorithm
Algorithm selection using model evaluation scale

Fig. 3 Model Creation Process

The process of model creation is as follows. First, data from a group of authorship of malware for the creation of models is collected. The groups of authorship of malware are APT 1, APT 10, APT 29, Gorgon Group, Lazarus Group, and Winnti. Second, the collected data is analyzed through automated analysis.

In the case of existing research, malware files have been converted into source code and binary code, but in this paper, an automated analysis tool is used for analysis without conversion. Third, Runtime Modules are extracted based on the results of the analysis. Fourth, the frequency analysis of each group is conducted, and the Module is selected according to a certain standard. Fifth, 6 Machine Learning classification algorithms are applied, and finally, the algorithm is selected by using Accuracy, Precision, Recall, and F1 Score, which are the measure of the evaluation of the artificial intelligence model. Fig. 4 is a method of identifying the authorship based on the proposed model.

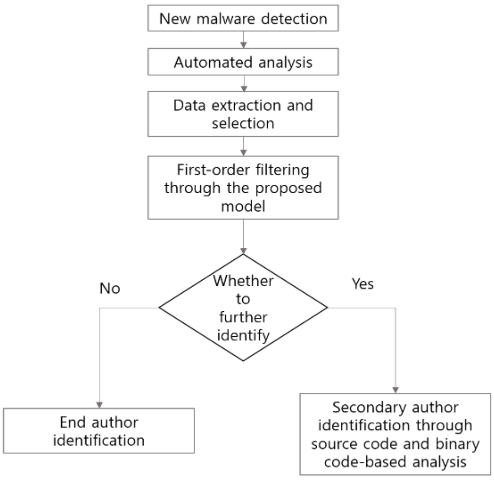

New malware detection
Automated analysis
Data extraction and selection
First-order filtering through the proposed model
Whether to further identify
No
Yes
End author identification
Secondary author identification through source code and binary code-based analysis

Fig. 4 Malware Authorship Attribution based on the Proposed Model

Methods of identifying the authorship of the new malware via the proposed model are as follows. First, when new malware is found, automated analysis is performed. Second, information from Runtime Modules is extracted through the results of analysis, and the necessary data is selected. Thirdly, the first process of attribution is based on the proposed model. If the results come out here, the attribution will be terminated, and if the result of the attribution in the automated analysis does not come out, or if there is a changed malware file other than the Windows execution file, the existing attribution method will be used, and the attribution will proceed.

F. *Selection of Runtime Modules*

This paper selected Runtime modules as a feature for attributing the authorship. Runtime modules are modules and dynamic libraries that are loaded into the Runtime when the file is run, and in the case of Windows execution files, the list can be extracted through automated analysis. In addition, in the case of Runtime Modules, Module, which each group of authorship of malware frequently uses, was selected as a feature of attribution. Table 3 is an example of the frequency analysis results of APT 1 and Lazarus Group. Ole32.dll was used at a frequency of 19% in APT 1, while Lazarus Group was used at a frequency of 60%. In the case of Wininet.dll, it was used at a frequency of 60% in APT 1, while in the case of Lazarus Group, it was used at a frequency of 16%. Therefore, in this study, the configuration conditions of the frequency analysis dataset were created, and the Module classified according to the conditions was composed of the dataset. The conditions for the configuration of the frequency analysis dataset are as follows.

- More than 40% chance of use in at least one group
- More than 30% difference in the frequency with other groups

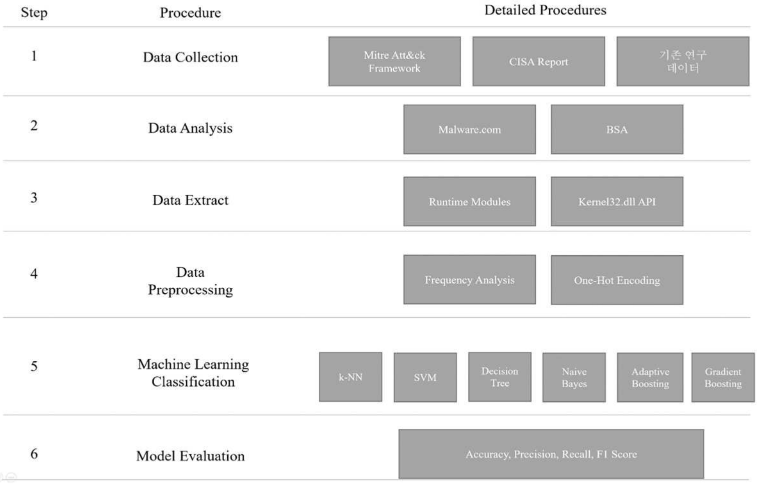
TABLE III
EXAMPLE OF FREQUENCY ANALYSIS RESULT

Runtime Modules	APT 1	Lazarus Group	Dataset
Advapi32.dll	90%	83%	X
Kernel32.dll	84%	93%	X
Ole32.dll	19%	60%	O
Msvcdrt.dll	9%	36%	X
Wininet.dll	60%	16%	O
Sspicli.dll	2%	43%	O

III. RESULTS AND DISCUSSION

A. *Experimental method*

The overall experimental method is shown in Fig. 5. The first step is the way the data is collected.



Step	Procedure	Detailed Procedures
1	Data Collection	Mitre Att&ck Framework CISA Report 기존 연구 데이터
2	Data Analysis	Malware.com BSA
3	Data Extract	Runtime Modules Kernel32.dll API
4	Data Preprocessing	Frequency Analysis One-Hot Encoding
5	Machine Learning Classification	k-NN SVM Decision Tree Naive Bayes Adaptive Boosting Gradient Boosting
6	Model Evaluation	Accuracy, Precision, Recall, F1 Score

Fig. 5 Experimental Method

Data is collected through the MITRE ATT&CK framework, the CISA report, and data from existing research. The data of the experiment is shown in Table 4.

TABLE IV
EXPERIMENTAL DATA

Runtime Modules	Number of dataset
APT 1	52
APT 10	53
Lazarus Group	30
Winnti	50
Gorgon Group	48
APT29	33
Total	266

The second step is the analysis of data. Automatic analysis tools from Malwares.com, and Buster Sandbox Analysis (BSA) is used for analysis. The third step is the extraction of data. A list of Runtime Modules based on Windows executable files is extracted. The fourth step is the preprocessing of the data. Frequency analysis allows the right data to be extracted and the dataset to be configured. After that, preprocessing is done through One-Hot Encoding.

The fifth step is the application of the Machine Learning classification algorithm, and six classification algorithms such as k-NN, and SVM are applied. The final sixth step is the stage in which the model is evaluated based on the results presented after the application of the classification algorithm. The analysis is carried out using four scales as equation (1)-(4). In Table 5, P and N means positive and negative.

TABLE V
CLASSIFICATION RESULT

		Real Result	
		True	False
Classification Result	True	TP	FP
	False	FN	TN

Among the measures of evaluation, accuracy is the simplest measure of performance, and the formula is obtained as follows.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+\ TN} \tag{1}$$

Precision is the proportion of real true data is measured among the results predicted by the model to be true, and the formula to be obtained is as follows.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

The recall is the proportion of data that the model is predicted to be true is measured among data that is actually true, and the formula is obtained as follows.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

F1 Score is the value of the harmonic mean of precision and recall, and the formula is obtained as follows.

$$F1\ \ Score = 2 \cdot \frac{Precision\ \cdot\ Recall}{Precision+Recall} \tag{4}$$

B. Experimental Result

The experimental results show the accuracy of attribution according to each classification algorithm from 2 to 6 groups of malware authorship. First, the k-NN algorithm shows in Table 6. Accuracy has increased compared to previous studies, but in the case of Group 6, it can be seen that there is a difference of more than 40% in the accuracy and the different evaluation scale.

TABLE VI
APPLICATION OF K-NN ALGORITHM

	2 Group	**3 Group**	**4 Group**	**5 Group**	**6 Group**
Accuracy	100%	97.5%	96.4%	90%	85%
Precision	94%	85%	64%	62%	44%
Recall	94%	75%	62%	56%	41%
F1 Score	94%	75%	62%	56%	41%

For the SVM algorithm, the results are shown in Table 7.

As 87% accuracy is seen in 6 group, it can be seen that the accuracy is higher than the previous study, and it can be seen that there is not much difference from the changed evaluation scale.

TABLE VII
APPLICATION OF SVM ALGORITHM

	2 Group	**3 Group**	**4 Group**	**5 Group**	**6 Group**
Accuracy	100%	96.2%	94.6%	89.3%	87%
Precision	100%	97%	95%	91%	88%
Recall	100%	97%	95%	90%	87%
F1 Score	100%	96%	95%	89%	87%

For the Decision Tree algorithm, the results are shown in Table 8. While 93.5% accuracy was shown in the 6 group, the difference from the changed evaluation scale was high at 10%.

TABLE VIII
APPLICATION OF DECISION TREE ALGORITHM

	2 Group	3 Group	4 Group	5 Group	6 Group
Accuracy	100%	100%	99.3%	94.8%	93.5%
Precision	96%	97%	92%	89%	88%
Recall	96%	97%	91%	88%	87%
F1 Score	96%	97%	91%	88%	87%

For the Naive Bayes algorithm, the results are shown in Table 9. In the 2 groups, 93.5% accuracy is shown, but as the number of authorship increases, the accuracy can be seen to be reduced, and precision, recall, and F1 Score can be seen to be lower than other machine learning algorithms.

TABLE IX
APPLICATION OF NAÏVE BAYES ALGORITHM

	2 Group	**3 Group**	**4 Group**	**5 Group**	**6 Group**
Accuracy	93.5%	92.5%	83.6%	71.4%	63.7%
Precision	94%	93%	87%	81%	71%
Recall	94%	93%	84%	71%	64%
F1 Score	94%	93%	84%	71%	62%

For the Adaptive Boosting algorithm, the results are shown in Table 10. 100% accuracy is shown when two groups are formed, but as with the Naive Bayes algorithm, the more authorship, the worse the overall performance is.

TABLE X
APPLICATION OF ADAPTIVE BOOSTING ALGORITHM

	2 Group	3 Group	4 Group	5 Group	6 Group
Accuracy	100%	96.2%	89.2%	65.9%	57.4%
Precision	100%	97%	91%	65%	66%
Recall	100%	96%	89%	66%	57%
F1 Score	100%	96%	89%	66%	57%

For the Gradient Boosting algorithm, the results are shown in Table 11. It can be seen that high accuracy is shown at 87.1% when the six groups are formed and that there is not much different from other evaluation scales.

TABLE XI
APPLICATION OF GRADIENT BOOSTING ALGORITHM

	2 Group	3 Group	4 Group	5 Group	6 Group
Accuracy	100%	96.2%	97.3%	91.5%	87.1%
Precision	100%	97%	98%	89%	89%
Recall	100%	96%	97%	85%	87%
F1 Score	100%	96%	97%	85%	87%

C. Analysis of Results and Comparison with Existing Research

In the case of k-NN and Decision Tree algorithms, the accuracy has been measured high, but the difference from the other evaluation scale has increased. In the case of Naive Bayes and Adaptive Boosting algorithms, as the number of the authorship group increases, the accuracy is significantly reduced, so it is not appropriate to identify the authorship of the malware. However, in the case of the SVM algorithm and the Gradient Boosting algorithm, the accuracy was highly measured, the difference from the other evaluation scale was insufficient, and when there were six authorship, more than 85% accuracy was shown, which was suitable for the attribution of the authorship of the malware, and the results of the experiment with the Gradient Boosting algorithm showed more accuracy and four better evaluation results than the SVM algorithm, so in this study, the results of experiments using Gradient Boosting algorithms were compared with previous studies.

The comparison was conducted in a framework in which Alrabaee et al. [14]'s OBA2 methodology and the characteristics of a group of malware attackers based on Shin et al. [4]'s genetic algorithm are extracted. First of all, the OBA2 methodology study shows 95% accuracy when authorship is 2, 90% when there are 4, and 84% when there are 6. The results of applying the Gradient Boosting algorithm through the single value of Runtime Modules proposed in this study have better attribution accuracy by seeing the accuracy of 100% of authorship in 2, 97.3% in 4, and 87.1% of authorship in 6. Table 12 shows this result.

TABLE XII
COMPARISON RESULT WITH ALRABAEE ET AL [14]

Number of Authors	OBA2 [14]	Proposed Model
2	95%	100%
4	90%	97.3%
6	84%	87.1%

In addition, the framework for extracting the characteristics of the attacker group based on the genetic algorithm showed 84% accuracy in the case of 5 authors, and the result of the application of the Gradient Boosting algorithm through the single value of the Runtime Modules suggested in this study was 91.5% when the authors were 5, which is 7.5%p higher than the previous study.

IV. CONCLUSION

We have proposed the malware authorship attribution model using runtime modules based on automated analysis. We have conducted to identify the authorship of malware that is produced due to recent standardization and modularization, and automated analysis to solve the time to identify the authorship of the existing malware and the limit of the workforce used. In addition, the Runtime Modules, which is the value of the results of the automated analysis, were frequently analyzed to attribute the authorship. Based on the selected features, 6 Machine Learning algorithms were applied to analyze the results, and accuracy comparisons with existing studies were conducted. As a result of the application of 6 algorithms based on the characteristics proposed in this study, the accuracy was highly measured in the SVM and Gradient Boosting algorithm, and the difference between precision, recall, and F1 score, a measure of evaluation of other algorithms, did not increase significantly. It is an algorithm appropriate to attribute the authorship of the malware, and the Gradient Boosting algorithm was chosen in this study because the Gradient Boosting algorithm has better overall performance than the SVM algorithm.

Compared to previous studies, accuracy was improved by up to 7.3%p, and high accuracy was also shown in a comparison of overall accuracy. Even at the stage of extracting features required to attribute the authorship of malware, existing research takes a long time as various analysis methods such as dynamic analysis and static analysis are used to extract source code-based characteristics or binary-based characteristics. On the other hand, the use of automated analysis in this study can reduce time-consuming in the process of extracting the characteristics of the authorship attribution of malware. Therefore, this study is differentiated from existing research because the authorship can be attributed with relatively high accuracy for malware produced due to standardization and modularization, and it is expected that the authorship will be attributed to new fast-generating malware and the damage will be reduced.

Future studies will be conducted to ensure that the limitations of automated analysis that does not produce the results of the analysis due to obfuscation or packing will be resolved, the characteristics of attribution will be visualized to improve accuracy, and Deep Learning will be used to attribute the authorship of the malware.

ACKNOWLEDGMENT

This research was supported by the 2022 scientific promotion program funded by Jeju National University. This article is excerpted from the presented Master's dissertation of the first author.

REFERENCES

[1] Kamundala Espoir K ,and Kim Chang Hoon. "CNN Model to Classify Malware Using Image Feature." KIISE Transactions on Computing Practices(KTCP), Vol. 24, No. 5, pp. 256-261, May. 2018

[2] Young-Bok Cho. "The Malware Detection Using Deep Learning based R-CNN." Journal of Digital Contents Society, Vol.19, No. 6, pp. 1177-1183, Jun.2018

[3] Ji-Won Hong, Sang-Hyun Park, Sang-Wook Kim. "Malware Feature Selection for Author Group Classification." KIISE Database Society of Korea, Vol. 34, No. 1, pp. 14-24, Apr. 2018

[4] Gun-Yoon Shin, Dong-Wook Kim, Myung-Mook Han. "The attacker group feature extraction framework: Authorship Clustering based on Genetic Algorithm for Malware Authorship Group Identification." Journal of Internet Computing and Services Vol. 21, No. 2, pp. 1-8, Apr. 2020.

[5] Saed Alrabaee, Paria Shirani, Mourad Debbabi, Lingyu Wang. “On the Feasibility of Malware Authorship Attribution." International Symposium on Foundations and Practice of Security, vol 10128, pp. 256-272, Dec. 2017

[6] E. Stamatatos. “A Survey of Modern Authorship Attribution Methods." American Society for Information Science and Technology Vol. 60, No. 3, pp. 538-556, Mar. 2009

[7] A. Caliskan-Islam, R. Harang, A. Liu, A. Narayanan, C. Voss, F. Yamaguchi. “De-anonymizing Programmers via Code Stylometry." 24th USENIX security Symposium Security 15, pp. 255-270, Aug. 2015

[8] Muhammad Ljaz, Muhammad Hanif Durad, Maliha Ismail. “Static and Dynamic Malware Analysis Using Machine Learning." Proceedings of 2019 16th International Bhurban Conference on Applied Sciences&Technology(IBCAST), pp. 687-691, Jan. 2019

[9] Gerard Biau, Luc Devroye. Lectures on the Nearest Neigbor Method. Springer Series in the Data Sciences. pp. 25-32, 2015

[10] S.V.N vishwanathan, M. Narasimha Murty. “SSVM : A Simple SVM Algorithm." Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290). pp. 2393-2398, Aug. 2002

[11] Antony J. Myles, Robert N. Feudale, Yang Liu, Nathanile A. Woody and Steven D. Brown. “An Introduction to Decision Tree Modeling." Journal of Chemometrics, Vol.18, Issue. 6, pp. 275-285, Jun. 2004

[12] Zhi-Hua Zhou. Ensemble Methods: Foundations and Algorithms. A Chapman & Hall Book. pp. 23-44, 2019

[13] Rosenblum, Nathan, Xiaojin Zhu, and Barton Miller. "Who wrote this code? identifying the authors of program binaries", ESORICS, pp. 172-189. 2011

[14] Saed Alrabaee, Noman Saleem, Stere Preda, Lingyu Wang, Mourad Debbabi. “OBA2: An Onion approach to Binary code Authorship Attribution." Digital Investigation, Vol. 11, Supplement. 1, pp. S94-S103, May. 2014

[15] Suk-Jin Hong, Ji-Won Hong, Sang-Wook Kim, Dong-Phil Kim, Won-ho Kim. "Malware Author Group Classification using Deep Learning Classifier." KIISE Database Society of Korea, Vol. 34, No. 2, pp. 34-45, Aug. 2018