Malware Classification Using Machine Learning

JinSu Kang and Yoojae Won(✉)

Department of Computer Science Engineering, Chungnam National University, Daejeon, South Korea
{kangjsl798,yjwon}@cnu.ac.kr

Abstract. Recently, tools for generating malware have been spreading rapidly on the internet, making it easier for people without expertise to create malware. As a result, the number of malware variants is increasing quickly. To address this issue, it is crucial to classify malware quickly and accurately. However, malware variants are evolving to evade traditional malware-detecting methods based on signature pattern matching. To solve this problem, researches on detection of malware have been made in various fields. In the present study, we first propose a classification method to extract feature data from malware files that is applicable to machine learning, and then we classify malware through learning. Finally, we apply our classification method to sample data to evaluate performance and analyze the results.

Keywords: Computer security · Metamorphic · Polymorphic · Malware classification · Machine learning

1 Introduction

Recently, tools for generating malware have been spreading rapidly on the internet, making it easier for people without expertise to create malware. As a result, the number of new malware variants is increasing quickly. According to AV-Test malware trend report, the number of recent malware has been increasing, but the number of new malware has not changed much. Most of the recent malwares are variant of existing malware [1]. Most of the new malware use either a polymorphic method for compressing and encrypting existing codes or a metamorphic method for transforming files to detour signature pattern matching-based malware detection employed in obtaining traditional anti-virus solutions. Currently, various malware analysis methods are being studied in response to the generation of malware variants. In the present study, we propose a malware classification method using machine learning, for which attribute data applicable to learning is extracted from malware data. This is followed by an analysis of the obtained results.

J. J. Park et al. (Eds.): CUTE 2018/CSA 2018, LNEE 536, pp. 279–284, 2020.
https://doi.org/10.1007/978-981-13-9341-9_48

2 Training Data Set

To carry out this study, we use the Microsoft Malware Classification Challenge data set (BIG 2015) hosted by Kaggle in 2015 [2]. For each data set, approximately 10,000 samples are provided in byte and disassembly file format to both the learning and the test data. Each data set is contained in one of the classes listed in Table 1.

Table 1. BIG 2015 data set classes

Number	Family name	Type
1	Ramnit	Worm
2	Lollipop	Adware
3	Kelihos_ver3	Backdoor
4	Vundo	Trojan
5	Simda	Backdoor
6	Tracur	Trojan downloader
7	Kelihos_ver1	Backdoor
8	Obfuscator.ACY	Obfuscated malware
9	Gatak	Backdoor

3 Feature Extraction

This section discusses the feature data extracted that is to be used for machine learning. In the present study, we extract the section name data of sequences, API, image data, and instruction and assembly code as features to use for learning. Each feature is explained in the following subsections.

3.1 Sequences

Malware files can be classified by analyzing byte sequences as they maintain binary format. For sequence analysis, the N-gram method is widely used. In the N-gram method, a long string is divided into multiple strings of size N, and then a statistical technique is employed to analyze the pattern of each fragment. In this study, we set N to be 4 and extract 4-gram data in the binary files of the malware sample data to use for learning [3].

3.2 Application Programming Interface (API)

API information regarding where a specific file is being imported can be identified by PE file analysis. API information is related to actual performance for PE file execution; thus, it is a key indicator for malware analysis. However, many recent malware variants exploit obfuscation; hence, API information to be extracted from files is limited. This makes it difficult to use API information alone for learning and requires its use in

conjunction with other features [4, 5]. In the present study, we measure the number of API calls in the malware sample files and store it to use as learning data (Fig. 1).

Fig. 1. API call extraction.

3.3 Section Name

The PE file is configured with pre-defined section names, including .text, .data, .idata, .edata, .rdata, .rsrc, .tls and .reloc [6]. Each section includes a code to execute a program, a global variable, and DLL information for importing and exporting resource-related data. However, the PE file is changeable because file characteristics are not mandated. While normal files tend to maintain the format of section names, malware uses an arbitrary section name for its obfuscation. Accordingly, section names are extracted as a feature to be used for learning. In our research, we extract the section names existing in the learning data and count the number of each file's section lines to use for learning.

3.4 Instruction

The binary file is difficult to intuitively understand and thus becomes tricky to analyze. For more effective analysis, the assembly code of a file needs to be considered. In fact, it is in the assembly code that instruction information can be found [7]. The sequence can be analyzed using the N-gram method as one does in a byte file. In this study, we measure the frequency of instruction to use it as feature data.

3.5 Image Representation

Because malware variants are visually identical to existing malware via imaging, recent studies have carried out the imaging of malware and then employed image-treating algorithms such as convolution neural networks (CNNs). Figure 2 shows the method for imaging malware. Imaging is performed by transforming each byte in the malware binary file into an 8-bit vector and subsequently turning it into malware, which is represented by the greyscale image in Fig. 2, with values between 0 and 255 [8].

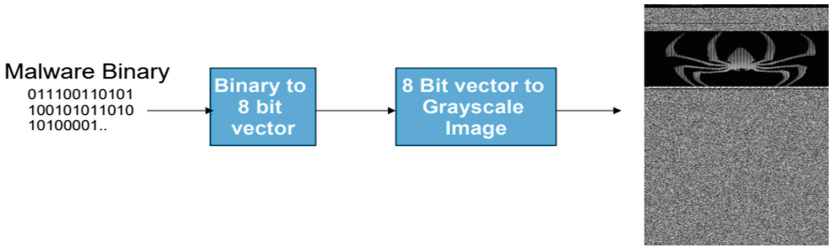

Fig. 2. Visualizing malware as an image.

Imaging is possible for both binary files and assembly files of malware sample data. In the present study, the images of the assembly files are clearer than those of the binary files, and thus they are used as features. Figure 3 shows the image pattern of each class after imaging malware learning data. In this study, we extract 8-bit vectors from the malware binary files and store them to use as learning data.

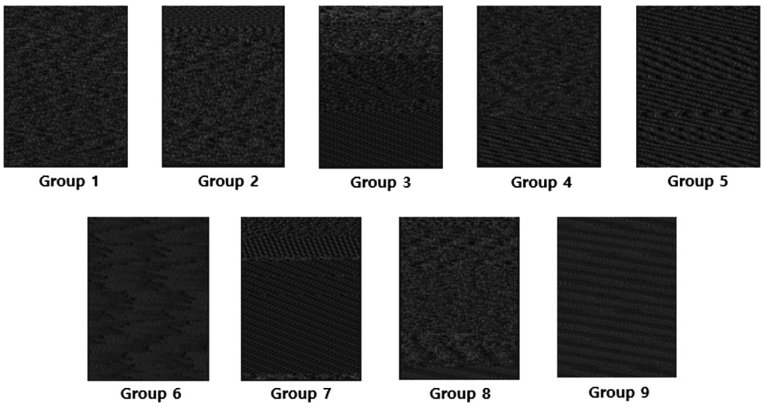

Fig. 3. Training a malware image.

4 Learning and Performance Evaluation

This section discusses the results of extracting the feature data discussed in Sect. 3 from malware sample data and employing machine learning. In the present study, we conduct learning using random forest, an ensemble technique-applied machine learning algorithm, followed by an assessment of the algorithm's performance. Random forest is an algorithm that creates multiple decision trees during training and performs classification and prediction of the mean value. In this study, 75% of the entire data is used as a training set for the experiment, and the remaining 25% is used as a test setup to assess performance. In addition, the out-of-bag (OOB) score is measured while applying the random forest to assess performance (Fig. 4).

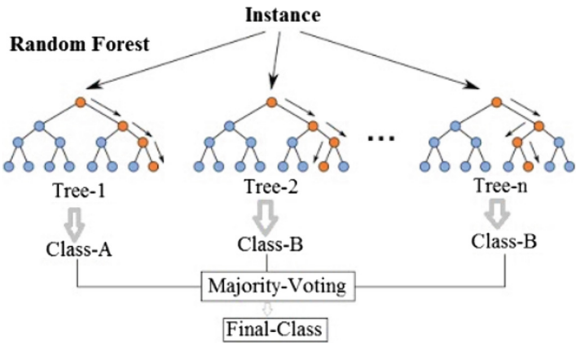

Fig. 4. Random forest simplified.

Table 2 shows measured performance. After training the model using the training set, the performance was measured using the test set, and the accuracy was measured as 99.8%. Likewise, we measured the performance using OOB data and achieved an accuracy of 99.5%.

Table 2. Evaluation result

Data	Accuracy
Test set	99.8%
Out of bag	99.5%

Figure 5 shows a confusion matrix. We have confirmed that most of the data are correctly classified.

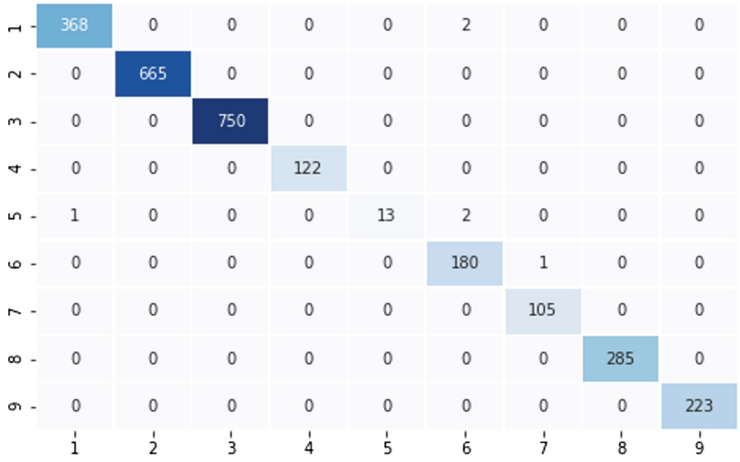

Fig. 5. Confusion matrix

5 Conclusion

Recently, many studies have been carried out to deal with the rapid emergence of malware variants. In the present study, we propose a method for classifying malware using machine learning and conduct related experiments. After performing the learning procedure using the Microsoft Malware Classification Challenge data set, we analyze its performance, and an accuracy of 99%. In future research, we will develop new methods for improving the performance of malware classification. In addition, we will explore how to develop a data set using both normal and malware files to analyze malware-detection performance.

Acknowledgments. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2016-0-00304) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

1. AV-Test: Security Report 2016–2017 (2016)
2. https://www.kaggle.com/c/malware-classification
3. Santos, I., Penya, Y., Devesa, J., Bringas, P.: N-grams-based file signatures for malware detection. In: Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS), AIDSS, pp. 317–320 (2009)
4. Islam, R., Tian, R., Ba, L.M., Versteeg, S.: Classification malware based on integrated static and dynamic features. J. Netw. Comput. Appl. **36**, 646–656 (2013)
5. Ki, Y., Kim, E., Kim, H.: A novel approach to detect malware based on API call sequence analysis. Int. J. Distrib. Sens. Netw. **11**(6), 4 (2015)
6. Ahmadi, M., Ulyanov, D., Semenov, S., Trofimov, M., Giacinto, G.: Novel feature extraction, selection and fusion for effective malware family classification. In: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy (2016)
7. Santos, I., Brezo, F., Nieves, J., Penya, Y., Sanz, B., Laorden, C., Bringas, P.: Idea: opcode-sequence based malware detection. In: Engineering Secure Software and Systems. LNCS, vol. 5965, pp. 35–43 (2010)
8. Nataraj, L., Karthikeyan, S., Jacob, G., Manjunath, B.S.: Malware images: visualization and automatic classification. In: Proceedings of the 4th ACM Workshops on Security and Artificial Intelligence, pp. 21–30 (2011)