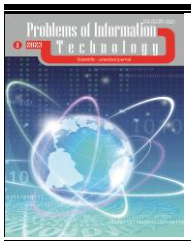

Available online at www.jpit.az

14 (1)
2023

Malware detection based on opcode frequency

Elshan O. Baghirov

Institute of Information Technology, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

elsenbagirov1995@gmail.com

ARTICLE INFO

http://doi.org/10.25045/jpit.v14.i1.01

Article history:
Received 6 September 2022
Received in revised form 9 November 2022
Accepted 23 December 2022

Keywords:
Opcode frequency
Correlation
Malware
Signature
Obfuscation

ABSTRACT

The amount of new malware has been continuously growing, and its threats are increasing rapidly. Developing new types of detection methods and thereby protecting computer systems from malicious programs has always been of interest to scientific researchers, individuals and organizations. In this work, several classification methods are applied on the dataset which is prepared on the basis of opcodes obtained from known malicious and benign program samples. Dependency between opcodes higher than 70% of total are removed to achieve more relevant results. The other main factors affecting the results of the methods are evaluated. Results prove that Random Forest classifier can classify suspicious programs with higher accuracy than others.

1. Introduction

The war between malware distributors and the massive crowd mobilized for malware detection continues. The reason cybercriminals use malware as a weapon is because of its destructive potential. This includes huge financial losses, disruption of critical services, and even human casualties in critical infrastructures such as SCADA.

The recent increase in malware occurs at the expense of previous examples. AV-Test Institute registers 450,000 new types of malware samples per day. The number of malware registered in 2022 has already surpassed the number of malware registered during the previous one. The most important factor causing this problem is that there exist many resources that even non-experts can prepare malware using obfuscation techniques such as polymorphism and metamorphism (Arzu et al., 2020).

In practice, the most common approach to malware detection is the use of commercial antivirus products, which primarily perform signature-based detection (Santos et al., 2013, Carlin et al., 2019). Although the method is fast and effective, it is not stable against malicious programs that applied evasion methods, which reduces its reliability. Thus, malware producers use obfuscation methods such as packaging, polymorphism, and metamorphism on existing malware to update the malware signature. Currently, one of the biggest problems in the fight against malicious programs is the detection of these types of programs.

Different methods and solutions have been proposed to detect malware. Malware was first suggested to detect by extracting a sequence of unique characters from the binary code prior to execution. Although the method was accurate, it was ineffective for unknown malware. Later, heuristically, extracting features like opcode frequency, opcode n-gram, and opcode graph

from malicious programs and applying data mining methods on the extracted features were proposed as a solution (Zhang et al., 2019). Since the method has an important role in detecting malicious programs applied obfuscation methods, research in this field has begun to be strengthened. A dynamic analysis method was proposed when there was a need to study the malicious program by executing it. With this method, researches on extracting and analyzing features of malicious programs in dynamic environments (sandbox, virtual machine) such as API requests, URL, system requests, opcodes were expanded. At the same time, since the dynamic analysis is quite time-consuming, there was a need to develop more advanced methods.

Opcode frequency-based detection of malware applied obfuscation techniques continues to be relevant. Because opcodes are one of the most predictive approaches for detecting obfuscated malware. So, it can get higher accuracy than the other techniques (Zhang et al., 2016). In this work, application of machine learning algorithms is performed on opcode frequency database created from opcodes statically obtained from malware. An overview and comparative analysis of the work carried out for the detection of malicious programs based on the opcode sequence is performed. On the basis of opcode samples obtained from malicious and benign programs, machine learning algorithms with default hyperparameters are trained, tested with relevant metrics and comparative analysis is carried out. How processing works on the base, as well as optimization of hyperparameters, affect the result is studied.

The 2nd section of the study provides information on the research conducted on the method of detection based on the opcodes obtained from malicious programs. The 3rd section describes practical information about the experiments carried out. The final section deals with the summary and future research.

Zhang et al. (2019) offered a new ransomware classification method based on opcode sequence prepared with static analysis. All architecture and workflow of the framework consists of 3 components like preprocessing, patched SA - CNN and bi-directional Self-Attention Network. The main drawback of the method is that it is difficult to process malware applied to the packaging method.

Khalilian et al. (2018) proposed Graph Mining for detection malware that applied metamorphism technique and used three classifiers, namely Decision Tree, Naïve Bayes, and Logistic Regression.

The main objective of the study is listed below:

- To analyze the current state of works done in the literature for the detection of malicious programs based on opcode frequency, to conduct a comparative analysis;
- To test machine-learning algorithms and determine the best algorithm through a dataset based on opcode frequency;
- To evaluate the impact of processing processes on the result of the algorithm on the set of data prepared based on opcode frequency.

2. Related works

A number of studies have been conducted to detect malicious programs based on opcodes.

The dataset developed based on the opcode frequency is collected from the literature, tested with 23 machine-learning algorithms with both default and optimized parameters, and a comparative analysis is carried out by Carlin (2019). SMOTE method is used to overcome the problem of oversample of minority class. Applying oversampling to dataset can increase the accuracy, but model will have difficulty to classify validation dataset.

McLaughlin et al. (2021) reviewed data multiplication methods based on opcode sequences for detecting malicious programs on large and small datasets, analyzed the results obtained by these methods, and proposed a data augmentation method. Data augmentation is used to increase the dataset by adding slightly modified copies of existing data or newly created synthetic data from existing data. But augmented samples may be highly correlated with the existing training data.

James et al. (2018) focused on crypto-ransomware affecting Windows OS only. Opcodes are extracted from malware and benign files statically and preprocessed in Excel. 8 various feature selection methods are measured for feature reduction. Machine learning models

are trained using Weka tool.

Renjie. (2019) proposes a new and efficient approach which can automatically learn the opcode sequence patterns of malware. Firstly, opcode sequence is extracted using disassembly tool IDA Pro. To learn the feature vector representation of opcode word embedding technique is used. Finally, a two-stage LSTM model is proposed for detection of malware.

Seungho and Jongsub (2020) proposed a method that learned parameters of a deep-learning model based on opcodes, without executing the program code. They also offered CRNN (Convolutional Recurrent Neural Network) model that used opcode sequence as input.

The above approaches are shown to work in specific cases and problems, but there is no universally agreed upon method that is generally useful.

3. Experiments

The experiments are shown in steps organized in the following order.

3.1. Dataset preparation

Samples of data set in "CSV" format with 2000 malware samples, 1000 benign file samples, 30 opcodes were taken from github. In dataset columns (features), rows and cells demonstrate opcode name, suspicious file (document), opcode frequency in document respectively. Opcodes are generated using IDAPro tool.

An opcode sequence is a textual form obtained by statically decompiling a given program or dynamically executing that program. After preliminary processing (removed rows and columns that are not important in the model) on the collected samples, a dataset consisting of 998 harmless, 1738 malicious software samples and 28 columns, including one class, remained. The next processing task is to check the correlation between the opcodes. Fig. 1 shows the correlation between the independent variables, that is, the opcodes, obtained using the Pearson correlation method.

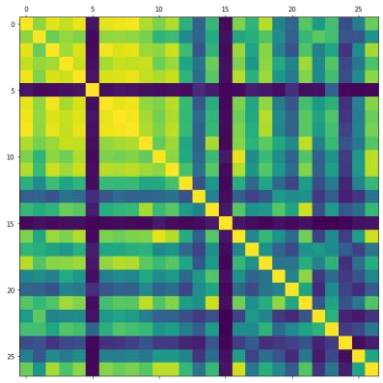

Fig. 1. Correlation between opcodes.

As can be seen from Figure 1, there is a high correlation (cells close to the color taken diagonally) between some opcodes. Those opcodes with a dependency between them higher than 70% are removed from the dataset and the learning process is updated.

The data set to be trained has two classes (malicious, benign) and the number of samples for these classes is approximately ½. Imbalance in the dataset is known to cause overfitting of the model. Two methods are often used to overcome this problem. The first is to equalize the number of classes by artificially increasing the data on the minority class (oversampling), and the second is to equalize the number of classes by reducing the data on the majority class (undersampling). Depending on factors, such as the amount of data used, the degree of risk, one of the two methods mentioned above can be chosen, or the parameters of the model can be changed to solve this problem without changing the data set.

One of the factors that can affect the result of the model is the feature importance. It is important not to consider features with a high feature importance in the data set or to assign weights to those features with a low value.

3.2. Training of classifiers

After preliminary processing on the data set, the training and test set are formed in the ratio of 2/8, the learning process is carried out using different classifiers, and the results obtained using the F-1 metric are compared. Target of dataset is distributed approximately 1:2 ratio (imbalanced). Using accuracy, precision, recall metrics separately can be misleading due to imbalance problem. For this reason, it would be the best option to use F1 score metric.

In cases where it is necessary to repeat the processing process on the features, returning to the previous stage, the classifiers are trained again after the corresponding processing.

The classifiers used are "Decision Tree", "Gradient Boosting", "KNN", "Logistic Regression", "Random Forest", "SGD", "SVM", "XGBoost" with default values.

3.3. Results

In order to increase the accuracy of the models, various hyperparameters corresponding to each model were considered, and the hyperparameters of significant importance were optimized through the "gridsearch" method and the models were retrained.

The experiments were conducted on the DataIKU platform. The obtained results are shown in table 1.

Table 1. Results of classifiers

Roc-Auc	1	0.99	0.99	1	0.99	0.99	0.99	***1***
F1	0.99	0.98	0.99	0.99	0.98	0.96	0.99	***1***
Recall	0.99	0.99	0.98	0.99	0.98	0.96	0.99	***0.99***
Precision	1	0.98	0.99	0.99	0.99	0.96	0.99	***1***
Accuracy	0.99	0.98	0.98	0.99	0.98	0.95	0.99	***0.99***
Classifier	Xgboost	KNN	Decision Tree	Gradient Boosting Tree	SGD	SVM	Logistic regression	***Random Forest***

The sequence of opcodes affecting the accuracy of the model is shown in fig. 2.

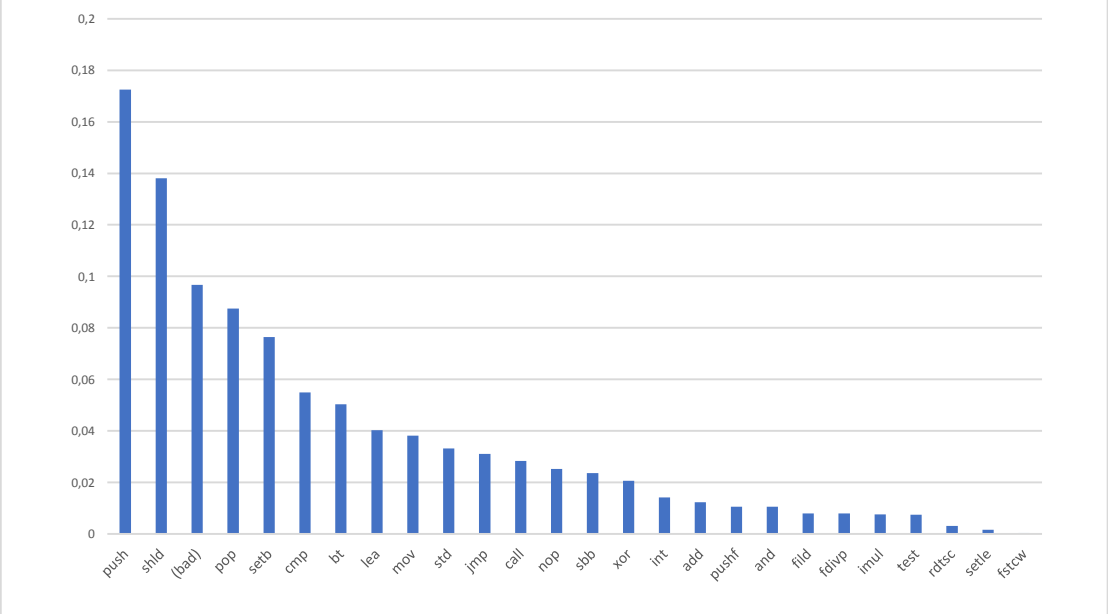

Fig. 2. Importance of opcodes in Random Forest classifer

Confusion matrix for Random Forest classifier (number_of_trees=100, depths=15, min_samples= 1) is shown in table 2. Optimal cut-off (0.375) was found by optimizing for F1 Score.

Table 2. Confusion matrix

	Predicted (1)	Predicted (0)
Actual (1)	370	2
Actual (0)	1	182

4. Conclusions and future works

In this study we used opcode dataset generated by disassembling the portable executable files. Extracted opcode sequence are used as feature during the classification process to identify unknown file. To get better result correlation between opcodes were checked and opcodes with higher correlation were dropped from dataset. Results showed that Random Forest classifier could classify malware with higher accuracy than others.

Opcodes generated with programs such as IDAPro may contain irrelevant opcode names, which may reduce accuracy when included in the model. For this reason, the application of text classification on opcodes can be successful and test size must be increased to achieve reliable performance estimation.

References

Arzu G.K., Mert N., Ibrahim S. "Metamorphic malware identification using engine-specific patterns based on co-opcode graphs", Computer Standards & Interfaces, vol. 71, pp. 1-12, 2020.

Carlin D., Philip O., Sezer S. "A Cost Analysis of Machine Learning Using Dynamic Runtime Opcodes for Malware Detection", Computers & Security, vol. 85, pp. 138-155, 2019.

James B., Dehghantanha A. "Leveraging Support Vector Machine for Opcode Density Based Detection of Crypto-Ransomware", Cyber Threat Intelligence, pp. 107-136, 2018.

Khalilian A., Nourazar A., Vahidi M. et al. "G3MD: Mining frequent opcode sub-graphs for metamorphic malware detection of existing families", Expert Systems With Applications, vol. 112, pp. 15–33, 2018.

McLaughlin N., Rincon J. M. "Data augmentation for opcode sequence based malware detection", arXiv:2106.11821v1, pp. 1-11, 2021.

Renjie L. "Malware Detection with LSTM using Opcode Language", arXiv:1906.04593v1, pp. 1-7, 2019.

Santos I., Brezo F., Ugarte X.P. et al. "Opcode sequences as representation of executables for data-mining-based unknown malware detection", Information Sciences, vol. 231, pp. 64-82, 2013.

Seungho J., Jongsub M. "Malware-Detection Method with a Convolutional Recurrent Neural Network Using Opcode Sequences", Information Sciences, vol. 535, pp. 1-15, 2020.

Zhang J., Zheng Q., Yin H. et al. "A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding", Computers & Security, vol. 84, pp. 376-392, 2019.

Zhang J., Zheng Q., Yin H. et al. "IRMD: Malware variant Detection using opcode Image Recognition", IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), pp. 1175-1180, Wuhan, China, 13-16 December, 2016.