

Malware Detection Using Decision Tree Algorithm Based on Memory Features Engineering

Adhitya Nugraha*¹, Junta Zeniarja²

^{1,2}Universitas Dian Nuswantoro, Jl. Imam Bonjol No. 207, Indonesia (+6224) 3517261

E-mail : adhitya@dsn.dinus.ac.id*¹, junta@dsn.dinus.ac.id²

*Corresponding author

Abstract – Malware is malicious software that can harm, manipulate, steal from victim's device system. Due to the diverse needs of using internet services, security threats are also increasingly difficult to detect. Now attackers are starting to develop malware that can change their own signature which is referred to as polymorphism. Therefore, improvements in the traditional approach to detecting the presence of malware are needed to be improved. One of the malware detection approaches, memory-based analysis technique has proven to be a powerful and effective analytical technique in studying malware behavior. In this study, the implementation of a Decision Tree-based classification algorithm was carried out to analyze the data set. Classifier model was created for the purpose of classifying malware based on memory features engineering. The result shows that the Decision Tree machine learning algorithm has been well performed with accuracy to 99.982%, a false positive rate equal to 0.1% and precision equal to 99.977%.

Keywords – malware detection, decision tree, memory analysis, machine learning

1. INTRODUCTION

Malware is malicious software intended to manipulate, steal or damage the victim's device system. Malware detection can generally be carried out by anti-virus software by matching it based on its signature such as MD5 or SHA1 hash values, static strings, file metadata, etc. However, nowadays attackers are starting to develop malware that can change their own signature. This feature of malware is referred to as polymorphism. This clearly makes it difficult for anti-virus software to detect malware based on its signature. Therefore, the development of detection techniques using a behavior-based approach is the right solution in finding polymorphic malware[1].

In malware analysis, malware classification becomes important to know how malware can infect computer devices, the level of risk they pose and how to fight the malware. In several research cases, memory-based analysis techniques have proven to be a powerful and effective analytical technique in studying malware behavior[2]–[5]. Some information regarding the presence of malware can be found through several examples of memory activity, such as active and terminated processes, active network connections, registry, Dynamic Link Libraries (DLL). Furthermore, memory analysis can also be used to detect processes in hooking techniques that commonly used by malware to imitate legitimate processes. Based on memory-based feature extraction, the memory analysis process can provide accurate information about the behavior, activity and characteristics of malware[6], [7].

Due to the high complexity and time consumption of manual detection methods, machine learning is used to be able to generate insights and knowledge from data automatically[8], [9]. Some uses of machine learning are aimed at figuring out which algorithm can make the fastest and most accurate assessments. Therefore, choosing a different algorithm according to the purpose and type of input will have a huge impact on the results of the system classification.

Shaiful and Aizaini[10], proposed method an improved decision tree algorithm to classify malware. Data set was collected by using Cuckoo Sandbox with no explanation further. The accuracy for classifying malware into its family are 93.3% on multi class and 94.6% on binary class. Kumar et al.[11], proposed a hybrid classification approach by combining C4.5 decision tree and Bayes classifier. JAVA programming was chosen in the implementation of this work and performance is evaluated based on several parameters such as classification accuracy, space and time complexity. Based on the results obtained, it is proven that the hybrid technique has better accuracy and performance compared to the traditional implementation of each algorithm. Faizal et al.[12], proposed a new model that can perform new feature extraction based on the ransomware data set. This technique successfully extracted fourteen feature vectors at runtime and analyzed the musing online machine learning algorithm stop redict the presence of ransomware. In this study, 78550 ransomware data sets consisting of malware and benign data. The trial was conducted by comparing the modified decision tree algorithm, random forest and Ada Boost. The results showed that the highest accuracy value was obtained by the decision tree algorithm with a value of 99.56%. Mosaddek et al.[13], proposed a robust approach for detecting the presence of android malware in data sets. The approach uses a selective feature step extracted using the Ranker search method and the Profit Ratio attribute evaluator. For advanced analysis, the authors applied the Decision Tree, Random Forest, and Random Tree algorithms to classify preprocessed data sets to be malware and benign. The highest accuracy with a value of 97.21% was obtained using the Random Forest classifier algorithm.

Based on the background described previously, the objectives of this paper are (i) to propose optimal malware detection using memory feature engineering, (ii) apply machine learning techniques based on decision tree algorithms, and (iii) present the results of the proposed approach based on analyzed data sets.

2. RESEARCH METHOD

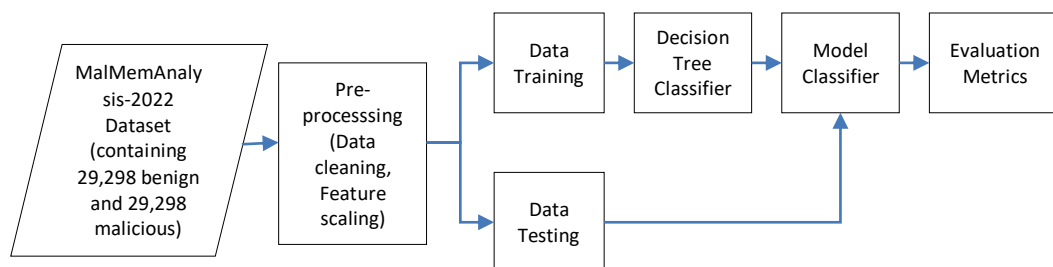


Figure 1. Proposed method

Fig. 1, presents a general scenario of the proposed approach to be carried out in this study. First, the selected data set is explored to find out the characteristics and features that exist. The second stage describes the data set preprocessing process to find and clean data

from incorrect or duplicate data. Features caling is applied to normalize the range of variables on the data features present in the data set. In the third stage, the data set is divided into testing and training sub sets, each of which will be used to build a classification and evaluation model. The final stage, for the purpose of classification a decision tree machine learning algorithm is used and evaluated.

2.1. Data Sets

The data set that used is MalMemAnalysis-2022[9] consist of 58,596 records with 29,298 benign and 29,298 malicious records. The dataset has a total of 57 features. The data set was made up of Spyware, Ransomware and Trojan Horse malware memory dumps process that taken using debug mode to avoid the unnecessary dumping process to show up in the memory dumps. (available data set in "<https://www.unb.ca/cic/datasets/MalMem-2022.html>").

2.2. Pre-processing

This phase perfoms preprocessing step to removes duplication data that make incorrect conclusions. Features caling also used to normalize the range of independent variables or features of data. Data available after the preprocessing is now appropriate for data modeling.

2.3. Decision Tree Classifier Model

In this phase, the implementation of the Decision Tree-based classification algorithm is carried out to train the classifier model. One of the important reasons for using a decision tree based algorithm is that it can perform classification without requiring a lot of computation. In addition, Decision Trees can provide a clear indication of which areas are most important for classification and generate understand able rules[10]–[12].

The preprocessed data set is divided into training and testing data sets with a ratio of 70:30. The training set is used to train the model, and the test set is used to validate the model results. The set training is used as input to the decision tree algorithm in the form of instance data. The results of this phase will form a training model which is the stage where machine learning takes place to study the input data set and then correlate the processed features to the sample output. This training model will later be used to run data testing through an algorithm to determine the accuracy of the model that has been formed.

2.4. Test set Input

After the model is trained, the next step is to validate it. For model validation, the test data set is processed in the same way as the training set. The results of the test calculations will be compared with the results of the training to calculate the accuracy of the model. Based on the calculation results against the test input set, if the predicted result is the same as the observed output, the classification accuracy increases otherwise the error will increase.

2.5. Evaluation Metric

Following performance metrics are calculated to determine the accuracy of our proposed model[13]:

$$a. Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$b. False Positive Rate = \frac{FP}{FP + TN} \quad (2)$$

$$c. Precision = \frac{TP}{TP + FP} \quad (3)$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

3. RESULTS AND DISCUSSION

This study aims to identify the behavior of malware through close observation of its features in memory. In this experiment, 55 features were used for analysis of the presence of malware in the data set. The effectiveness of memory-based features in malware detection and classification was validated by conducting this experiment. The experiment result shows that the Decision Tree machine learning algorithm has been well performed with accuracy to 99.982%, a false positive rate equal to 0.1% and precision equal to 99.977%. This indicates that the Decision Tree model is very good at identifying and detecting malware behavior.

In this research, gini importance value of each feature are calculated and features are ranked in descending order to shortlist the most important features. It is proof that several features such as `assvcscan.nservices`, `handles.avg_handles_per_proc`, `svcsan.process_services`, `malfind.commitCharge`, `svcsan.shared_process_services`, `handles.avg_handles_per_proc`, `svcsan.nservices`, `dlllist.ndlls`, and `pslist.nprocare` are the most important features in identifying malware behavior.

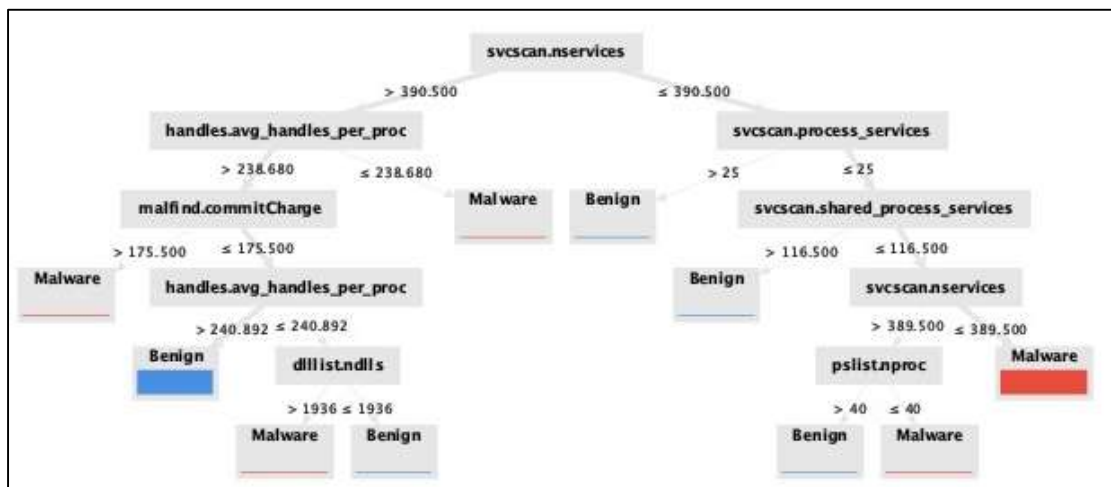


Figure 2. The Most Important Features

4. CONCLUSION

The purpose of this study is to develop a malware detection model using memory features to improve detection accuracy and reduce false positive rates. Memory analysis is gaining prominence for its ability to capture malware behavior. In this study, the classifier model by applying was developed using a data set consisting of 58,596 records with 29,298 benign data and 29,298 malicious data where the data set has a total of 57 features. The classification features in this model are also sorted using an extra tree classifier to select the best feature from the extracted feature list with the aim of improving processing performance.

Based on the results of the tests that have been carried out, it is obtained Decision Tree malware classification results achieve an accuracy rate to 99.982%, a false positive rate equal to 0.1% and precision equal to 99.977%. This indicates that the Decision Tree model is very good at identifying and detecting malware behavior.

REFERENCES

- [1] R. Sihwail, K. Omar, and K. A. Zainol Ariffin, "A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 4–2, pp. 1662–1671, 2018.
- [2] R. Sihwail, K. Omar, and K. A. Z. Ariffin, "An Effective Memory Analysis for Malware Detection and Classification," *Comput. Mater. Contin.*, vol. 67, no. 2, pp. 2301–2320, 2021.
- [3] S. Banin and G. Olav Dyrkolbotn, "Detection of Previously Unseen Malware using Memory Access Patterns Recorded before the Entry Point," *Proc. - 2020 IEEE Int. Conf. Big Data, Big Data 2020*, pp. 2242–2253, 2020.
- [4] A. H. Lashkari, B. Li, T. L. Carrier, and G. Kaur, "VolMemLyzer: Volatile Memory Analyzer for Malware Classification using Feature Engineering," *2021 Reconciling Data Anal. Autom. Privacy, Secur. A Big Data Challenge, RDAAPS 2021*, no. Cic, 2021.
- [5] B. Khilosiya and K. Makadiya, "Malware Analysis and Detection Using Memory Forensic," *Multidiscip. Int. Res. J. Gujarat Technol. Univ.*, vol. 2, no. 2, p. 106, 2020.
- [6] A. Singh, R. Ikuesan, and H. Venter, "Ransomware Detection using Process Memory," *Int. Conf. Cyber Warf. Secur.*, vol. 17, no. 1, pp. 413–422, 2022, doi: 10.34190/iccws.17.1.53.
- [7] Y. Gao, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Malware Detection Using Gradient Boosting Decision Trees with Customized Log Loss Function," in *International Conference on Information Networking*, 2021, vol. 2021-Janua, pp. 273–278.
- [8] R. Sihwail, K. Omar, K. A. Z. Ariffin, and S. Al Afghani, "Malware detection approach based on artifacts in memory image and dynamic analysis," *Appl. Sci.*, vol. 9, no. 18, 2019.
- [9] T. Carrier, P. Victor, A. Tekeoglu, and A. Lashkari, "Detecting Obfuscated Malware using Memory Feature Engineering," no. Icissp, pp. 177–188, 2022.
- [10] M. S. A. B. M. Sari and M. A. Maarof, "Classification of Malware Family Using Decision Tree Algorithm Phase : Features Identification and Classification," in *UTM Computing Proceedings: Innovations in Computing Technology and Applications*, 2017, vol. 2, no. 1, pp. 1–8.
- [11] A. Kumar, S. S. Singh, K. Singh, H. K. Shakya, and B. Biswas, *An Implementation of Malware Detection System Using Hybrid C4.5 Decision Tree Algorithm*, vol. 956, no. January. Springer Singapore, 2019.
- [12] F. Ullah *et al.*, "Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls," *Sci. Program.*, vol. 2020, 2020.
- [13] M. Hossain, S. Rafi, and S. Hossain, "An Optimized Decision Tree based Android Malware Detection Approach using Machine Learning," in *2020 7th International Conference on Smart Structures and Systems, ICSSS 2020*, 2020, pp. 117–125.