International Journal of Innovative Research in Engineering & Management (IJIREM)
ISSN: 2350-0557, Volume-9, Issue-3, June 2022
https://doi.org/10.55524/ijirem.2022.9.3.6
Article ID IJIR-1207, Pages 41-44
www.ijirem.org

Malware Recognition Using Machine Learning Methods Based on Semantic Behaviors

Praveen Hugar[1], Mayur Pershad[2], T.Sathvika[3], and Ganesh Bhukya[4]

[1]Assistant Professor, Department of Information Technology, J B Institute of Engineering and Technology, Moinabad, India
[2,3,4]Students, Department of Information Technology, J B Institute of Engineering and Technology, Moinabad, India

Correspondence should be addressed to Mayur Pershad; Mayur.pershads@gmail.com

ABSTRACT- Malware is any programme that gains access to or instals itself on a computer without the permission of the system's administrators. For cyber-criminals to achieve their nefarious objectives and purposes, a variety of viruses has been widely deployed. To tackle the growing number of malicious programmes and lessen their hazard, a novel deep learning framework is developed that employs NLP approaches as a starting point and combines CNN and LSTM neurones to record locally spatial correlations and learn from sequential longterm dependencies. As a result, for the malware classification job, high-level abstractions and representations are automatically derived. The accuracy of categorization rises from 0.81 (best by Random Forest) to about 1.0.

KEYWORDS- Machine Learning, Computer Security, Malware Recognition.

I. INTRODUCTION

Operating systems and the Internet have been subjected to a massive number of malicious applications that have created serious and developing security dangers. The number of new malware types has grown at an unprecedented rate, fueled by the significant revenues generated by cybercrime. According to Panda Labs, nearly 18 million new malware cases were discovered in the third quarter of 2016, averaging 200,000 samples each day. Automatic and effective malware detection and classification systems, which play critical roles in preserving the security of operating systems and networks, are critical in combating large-scale malware in the wild.

However, there are issues we are facing in terms of designing an effective and efficient process for malware detection and categorization. Malware authors have used a variety of concealment techniques to avoid detection, including encryption, packing, obfuscation, polymorphism, and metamorphism. Because the pattern matching method, which analyses the disassembly code statically and is readily beaten by standard evasion strategies, is still the most popular way for detecting suspected malware, there is a greater demand for better malware detection tools..

Machine learning and data mining have been used to address cybersecurity concerns such as detecting new and unexpected threats, and have shown to be successful in intrusion detection and malware categorization.

On malware identification and categorization, we used cutting-edge machine learning algorithms. With the goal of studying the performances of machine learning algorithms and uncovering statistical aspects of malware behaviors, seven classifiers and three feature representations were developed, compared, and assessed.

II. RELATED WORK

Over years, some researchers have integrated static and dynamic analysis to improve the accuracy and efficiency of automatic malware detection and classification. Wang proposed a surveillance spyware detection system in [7] by employing static features, including DLLs and API calls, and dynamic features, including modifications upon system files, network activities and registries. SVM was used asa classifier and the system reached 96.43% accuracy by using 10-fold cross-validation. Santos presented a hybrid malware detector in [6].

Static features were operational code sequences and dynamic features were extracted by monitoring operations, system calls, and raised exceptions. Results on 1000 malware and 1000 benign files demonstrated hybrid method strengthened the classification performance.

In [4], Islam combined function length frequency, printable sting, API function names along with the parameters to differentiated malware and benign files by hybrid analysis. In general, hybrid analysis combines advantages with both static and dynamic methods, and greatly enhances the understanding of malicious behaviours, and consequently, may lower the false positive rates. After representative features are prepared, classification is performed to categorise malware into different groups according to their similarities. Malware detection also can be regarded as s kind of binary classification.

III. HELPFUL HINTS

A. Figures and Tables

Figure 1 depicts the complete process of our effort. The key to detecting malware is to have a thorough grasp of its semantic features, which can capture real-world behaviours and represent dangerous functionality. Malware that belongs to the same category or family may exhibit similar behaviours, such as how to install on and infect OS systems, how to spread over networks, and how to get access to fundamental resources such as file systems, processes, threads, and devices.

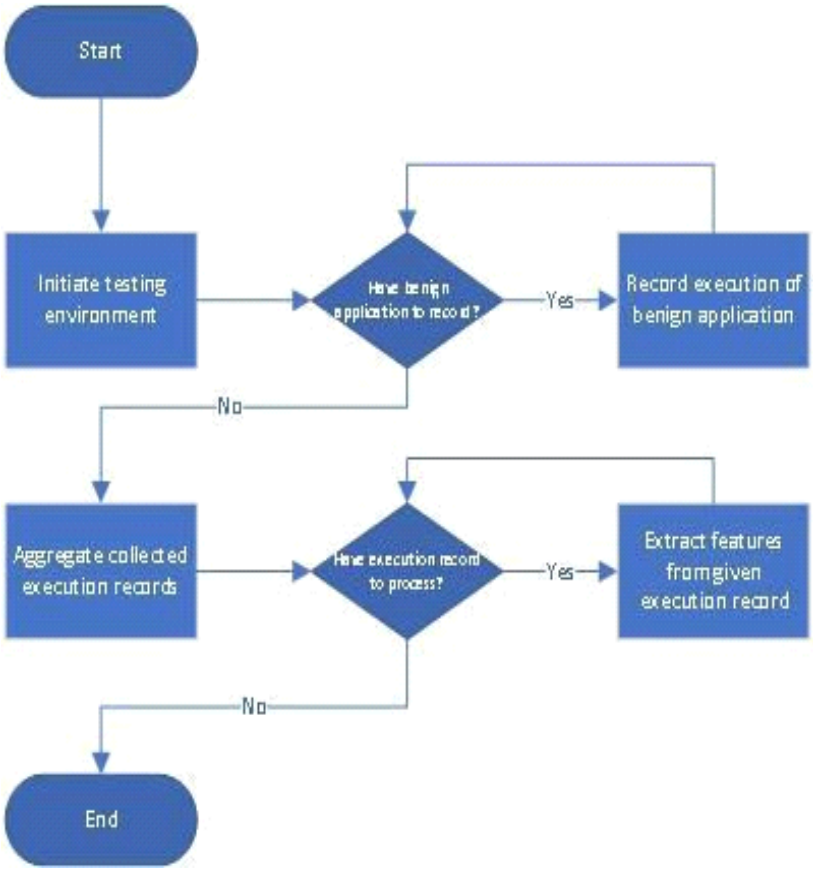

Figure. 1: Pervasive wireless grid

Three feature representations based on API call sequences are created and transformed to feature vectors to represent malware samples for further analysis in this paper. These representations comprise the frequency with which each API is invoked, the number of ngram API calls, and the Markov Chain transition matrix.

Then, in the framework of supervised learning, we apply numerous machine learning algorithms to malware classification, inspect, and assess their performance. Furthermore, these machine learning classifiers were applied to three different representations of malicious programmes based on the API calls sequence in order to reveal the statistical properties of malware behaviours and gain a better understanding of semantic characteristics from the surface to what lies beneath.

Multinomial logistic regression (MLR), k-nearest neighbours (kNN), Decision Tree(DT), Random Forest(RF), SVM, Naive Bayes(NB), and Multilayer Perceptron are among the classifiers studied (MLP). Furthermore, Information Gain and Random Projection are used to choose and extract partial features, respectively, as an essential aspect of machine learning to minimise the high dimension of features and extract the most important qualities for classification.

A. Charecterisation And Feature Representations

Because API calls sequence was chosen as the semantic characteristic to define and characterise programme behaviour, three feature representations based on API calls sequence were created and transformed to feature vectors to represent malware samples for analysis. These representations include the frequency with which each API is invoked, n-gram API calls, and the Markov Chain transition matrix..

- Representation 1: API Call Invocation Frequency:

Based on the idea that malware instances from the same category or family may behave similarly, some vulnerable APIs may be used more frequently than those from other classes. As a result, the initial representation of API call characteristics being investigated is the frequency with which API calls are invoked across a program's whole sequence. Counting the incidence of all API calls for each sequence and integrating these frequencies into a fixed-length feature-vector for all programmes, using each sequence and the API Calls List as input. If a certain API appears continuously more than once, we simply measure the frequency at one to avoid code obfuscation by adding trash calls.

- Representation 2: Markov Chain Transition Matrix:

The transition matrix of Markov Chain, which reflects the likelihood of changing from one state to another in a stochastic process, is the last representation reviewed.

The Markov Chain may be thought of as a directed graph G = V,E >, with the vertices (set of V) representing API calls and the edges (set of E) representing transition probabilities from one call to the next in the series of API calls. An adjacent matrix built by all transition probabilities TNN may be used to depict the Markov Chain Graph of any application given N API calls in the API Calls List. The transition probability from one elementTij to the next is indicated by TNN for each elementTij in the adjacent matrix.

B. Feature selection and extraction

According to the dataset's statistical fact, there are a total of 902 distinct API calls. When translating the above-mentioned representations into feature vectors for each malware sample, the large number of different API calls may result in a feature space with a high and sparse dimensionality, resulting in the "Curse of Dimensionality."

C. Machine Learning Classifiers

Machine learning and data mining have been used to address cybersecurity issues such as discovering unknown and unforeseen attacks, and they have shown effectiveness in intrusion detection and malware classification [2], [3]. Several techniques are developed on the above described feature representations in the supervised learning context with the goal of analysing the effectiveness of typical machine learning classifiers when applied to the malware classification challenge.

IV. EXPERIMENTS

A) Accuracy:

The “accuracy” is used as the overall assessment for the performance of various classifiers. Mathematically, it is the proportion between the number of correct predictions and the total number of predictions made. Conceptually, accuracy reflects the degree of conformity and correctness of classifiers when compared to a true or absolute value.

$$\text{Accuracy} = \text{TP+TN/TP+TN+FP+FN}$$

B) Experiments Design

To create the input dataset, we extracted 17230 tagged malware samples from the APIMDS-dataset, together with their entire list of API calls sequence (categories with less than 10 examples were removed). The "Feature Representation" step generates feature vectors for all sequences based on three possible representations. Following that, Information Gain is used to identify the most important features (as in n-gram API representations), and Random Projection is used to reduce dimensionality. In the supervised learning scenario, seven machine learning classifiers are learned next.

To provide a more compelling evaluation of the prediction models, stratified 10-fold cross validation (CV) is employed to determine how well the classifiers under consideration will perform in practise. As a model assessment approach, stratified 10-fold CV divides the original dataset into ten subgroups at random, ensuring that each subset has about the same percentage of samples from each target class as the entire set.

One of the 10 sets is utilised for testing, while the other nine are mixed and used as a training set. Stratified 10-fold cross validation has the benefit over repeated subsampling in that all observations are utilised for both training and validation, and each observation is only used for validation once.

C) Experiments result and performance evaluation

1) Classification Accuracy of Different Classifiers and Representations: The classification accuracies of seven examined classifiers on three different feature representations are listed from Table I to Table III (Note: for each representation, results in red indicate the best result produced by the corresponding classifier; for each classifier, result in bold indicates which feature representation achieves the best performance).

In terms of the computational efficiency, kNN spends most of the time on computing the distance between the tested sample with all the rests to make a decision. SVM is a memoryintensive and time-consuming algorithm which requires a lot of time spending on parameters tuning, such as selecting the suitable kernel, regularisation penalties and . Comparatively, random forest can achieve a higher (or at least similar) accuracy as SVM but performs more efficiently and scalably than SVM. However, Both SVM and Random Forest are non-parametric methods, which means the complexity of algorithms will increase with the enlargement of training data set. As to Multilayer Perceptron, the computational time and memory usage will be increased as the additional hidden layers and units. Except for the listed accuracy, the precision-recall curves of these classifiers also plotted in Figure 2, which can demonstrate the comparison of performances intuitively. Under the circumstance of imbalanced data distribution, precision-recall curve provides more information than receiver operating characteristic (ROC) curve, the lager area under the curve, the more accurate classifier. Using the representation of invocation frequency of APIs as the illustration, it is obvious to find that Random Forest and Multi-layer Perceptron achieve better performance than others.

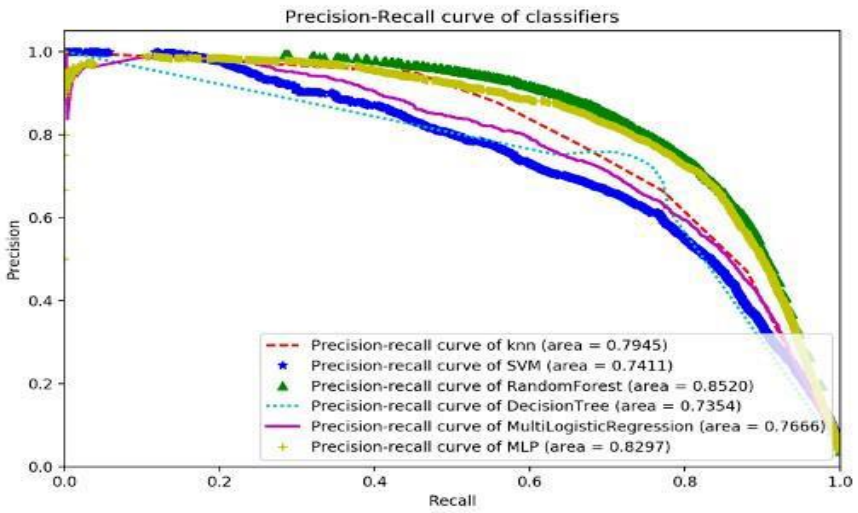

Figure 2 : Accuracy, the precision-recall curves of these classifiers also plotted in

V. CONCLUSION AND FUTURE ENHANCEMENTS

To get a greater classification accuracy when utilising typical machine learning algorithms, a rigorous feature engineering approach is required to choose and extract the most valuable features from the vast feature space.

This can be made more accurate with adding more data set. More algorithms with better performance can add on to accuracy. It can be hosted on web for real time analysis of exe files on the cloud

REFERENCES

[1] Nai Ding, Lucia Melloni, Xing Tian, and David Poeppel. Rulebased and word-level statistics-based processing of language: insights from neuroscience. Language, Cognition and Neuroscience, 32(5):570–575, 2017.

[2] Manuel Egele, Theodoor Scholte, Engin Kirda, and Christopher Kruegel. A survey on automated dynamic malware-analysis techniques and tools. ACM computing surveys (CSUR), 44(2):6, 2012.

[3] Ekta Gandotra, Divya Bansal, and Sanjeev Sofat. Malware analysis and classification: A survey. Journal of Information Security, 5(02):56, 2014.

[4] Rafiqul Islam, Ronghua Tian, Lynn M Batten, and Steve Versteeg. Classification of malware based on integrated static and dynamic features. Journal of Network and Computer Applications, 36(2):646–656, 2013.

[5] Judith Klein-Seetharaman Madhavi Ganapathiraju, Vijayalaxmi Manoharan. Blmt - statistical sequence analysis using n-grams. Applied Bioinformatics, 3(2-3):193–200, 2004.

[6] Igor Santos, Jaime Devesa, Felix Brezo, Javier Nieves, and Pablo Garcia Bringas. Opem: A static-dynamic approach for machine-learning-based malware detection. In International Joint Conference CISIS'12-ICEUTE 12-SOCO 12 Special Sessions, pages 271–280. Springer, 2013.

[7] Tzu-Yen Wang, Shi-Jinn Horng, Ming-Yang Su, Chin-Hsiung Wu, PengChu Wang, and Wei-Zen Su. A surveillance spyware detection system based on data mining methods. In Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, pages 3236– 3241. IEEE, 2006.