NLP- and API-Sequence-Based Malware Detection and Classification Methodologies

Subjects: Computer Science, Artificial Intelligence

Contributor: Peng Wang , Tongcan Lin , Di Wu , Jiacheng Zhu , Junfeng Wang

The surge in malware threats propelled by the rapid evolution of the internet and smart device technology necessitates effective automatic malware classification for robust system security.

Transformer malware classification API call sequences

1. Introduction

Malware, or malicious software, is crafted to infiltrate computers and mobile devices, aiming to manipulate authoritative systems, gather sensitive information, display unwanted ads, or extort users [1][2]. The surge in smart devices like laptops and phones has greatly expanded the threat landscape, jeopardizing user security and system integrity [3][4]. Malware classification assigns specific labels to identify its family, which is a crucial step in addressing security challenges [5].

Malware classification can be divided into signature-based, machine learning-based, and deep learning-based methods in the method view or static analysis and dynamic analysis in the feature view. Signature-based approaches may encounter challenges when dealing with the rapid evolution of malware [6]. In response, traditional machine learning methods, including Support Vector Machines (SVM), Random Forests (RF), and Naïve Bayes (NB), have been utilized for malware detection and classification [7][8]. However, these approaches necessitate the manual extraction of features, relying on expert knowledge, which can introduce complexity to the process.

Contemporary malware classification methods effectively leverage malware features, encompassing both static and dynamic attributes, to build machine learning or deep learning models. Static analysis involves the extraction of features as hex values and opcodes [9] from malware binary executable files through reverse engineering and examination of the original binary code. While static analysis is efficient, it is susceptible to evasion and obfuscation techniques. In contrast, dynamic analysis techniques capture malware behaviors, including file access, API (Application Programming Interface) calls, data flow, and other behavior traces, by executing and monitoring malware within a virtual sandbox. Dynamic analysis offers a more accurate representation of malware's actual objectives and actions, resulting in lower false-positive rates and higher accuracy [10][11]. Combined with deep learning's image representation, many research works would treat the malware as an image by converting the feature from static and dynamic analysis into a matrix [12][13].

Despite the success of feature analysis and deep learning, especially in image representation, the researchers posit that API call sequences can be regarded as a form of language through which programs establish communication with operating systems, analogous to how individuals employ languages for interpersonal interaction, which can better reflect the nature of the malware. Tran et al. point out that every type of malware has its own specific API call patterns or unique order of API calls [14]. In contrast to dynamic instruction features, the extraction of API call features necessitates only a coarse-grained dynamic analysis. Consequently, this approach incurs a relatively modest computational cost, rendering it highly effective for a broad spectrum of software codes.

2. Deep Learning-Based or API-Call-Related Malware Classification

There is a line of work focused on building malware classification systems based on extracted features. Nagano et al. [15] have proposed an innovative static analysis approach, integrating Natural Language Processing (NLP) with machine learning classifiers to discriminate between malicious and benign software. Their methodology entails the utilization of a PV-DBOW model for the extraction of features from diverse sources, including DLL imports, assembly code, and hex dumps, all derived from static analysis. Subsequently, these extracted features, or vectors, are input into Support Vector Machines (SVM) and k-nearest neighbor (KNN) classifiers for predictive inference. Another study proposed by Tran et al. [14] used NLP techniques such as N-gram, Doc2Vec (or paragraph vectors), and TF-IDF to convert API call sequences to numeric vectors before feeding them to the classifiers, including SVM, KNN, MLP, and RF. Schofield [16] also uses N-gram and TF-IDF to encode the API call sequences and employs a CNN to classify, which utilizes the ability of image representation. Chandrasekar Ravi et al. [17] employ a third-order Markov chain to model the Windows API call sequences. Nakazato J et al. [18] classify malware into some clusters using characteristics of the behavior, which are derived from Windows API calls in parallel threads with N-gram and TF-IDF.

Deep learning-based methodologies have exhibited remarkable potential for delivering more efficacious and adaptable features, yielding superior outcomes in malware classification. Kolosnjaji et al. [19] pioneered the application of convolutional and recurrent network layers for the extraction of features from comprehensive API sequences. Their pioneering work underscores the substantial accomplishments attained through the integration of deep learning techniques within API-sequence-based malware classification. In the same way, C Li's work [20] also demonstrates the RNN's ability to classify the API call sequences alone. In a subsequent development, Li et al. [21] have further refined the network architecture, introducing the extraction of inherent features from API sequences. Especially, their approach incorporates embedding layers to represent API phrases and semantic chains, along with the utilization of Bidirectional Long Short-Term Memory (Bi-LSTM) units to capture interrelationships among APIs. The results of their endeavors demonstrate significant performance enhancements when compared to baseline methodologies, highlighting the efficacy of introducing additional intrinsic features associated with APIs. Some works consider the similarity among the features, especially API call sequences, and employ similarity to do the encoder, followed by some advanced models such as GNN [22], Random Forest, LSTM [23], and F-RCNN [24].

3. Transformer Models and Local Attention

Transformer is the first sequence transduction model that relies entirely on the attention mechanism. Unlike RNN [25] and LSTM [26], Transformer [27] uses multi-headed self-attention instead of recurrent layers in encoder-decoder architecture. Thanks to the absence of recurrent layers, the Transformer does not need to face the risk of gradient disappearance and gradient explosion, and it can process the entire sequence and learn the relationship between API calls. Using the Transformer Encoder–Decoder model takes less time to train than the LSTM model, and it is more stable [28]. MalBERT [29] first utilizes the pre-trained Transformer to process and detect malware, and experiments demonstrate that the Bert-based model can achieve high accuracy for malware classification.

Transformer architecture delivers a good design of attention mechanisms; some work employs another attention module to capture the information. Yang [30] proposes to capture features from binary files using stacked CNNs and assembly files via triangular attention and then fuse all features via cross-attention. Their experimental results show that the method can extract both global and local features to improve the detection of malware variants effectively. Moreover, the local attention mechanism is very popular and effective in processing local features. Ma [31] points out that the mutual result of both global and local attention is useful to capture semantics and generate the most informative and discriminative features for text classification.

4. Training Strategies

Generally, benefiting from sufficient data, convolutional networks are always trained offline. Thus, researchers favor taking advantage of and developing better training methods that can not only promote the performance of the model but also have no inference cost increase. Inspired by [32], the researchers call this kind of method a “bag of freebies”. Strategies like data augmentation [33], hard negative example mining [34], online hard example mining [35], two-stage object detectors, and objective function designing [36], to name a few, are commonly used in computer vision and natural language processing (NLP).

In malware classification, Hwang [37] designs a two-stage detection method to protect the victims by employing random forest to control false negative error rates in the second stage under low false positive rates delivered by the first stage using the Markov chain model. Baek [38] employs static analysis and dynamic analysis in different stages; static analysis in the first stage is used to classify malware and benign files. After that, they further employ dynamic analysis in the second stage to classify malware from the benign files in stage one to lower the false detection rate and reduce the malware misclassification in stage one. The results show that a two-stage scheme can perform better than a single static analysis or dynamic analysis. Although these strategies can better improve the detection rate, current research lacks consideration of the representation of malware and detection speed performance.

References

1. Aboaoja, F.A.; Zainal, A.; Ghaleb, F.A.; Al-rimy, B.A.S.; Eisa, T.A.E.; Elnour, A.A.H. Malware Detection Issues, Challenges, and Future Directions: A Survey. Appl. Sci. 2022, 12, 8482.
2. Begovic, K.; Al-Ali, A.; Malluhi, Q. Cryptographic Ransomware Encryption Detection: Survey. Comput. Security 2023, 132, 103349.
3. Molloy, C.; Banks, J.; Ding, H.S.; Charland, P.; Walenstein, A.; Li, L. Adversarial Variational Modality Reconstruction and Regularization for Zero-Day Malware Variants Similarity Detection. In Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM), Orlando, FL, USA, 28 November–1 December 2022; pp. 1131–1136.
4. Ling, X.; Wu, L.; Zhang, J.; Qu, Z.; Deng, W.; Chen, X.; Qian, Y.; Wu, C.; Ji, S.; Luo, T. Adversarial Attacks against Windows PE Malware Detection: A Survey of the State-of-the-Art. Comput. Secur. 2023, 128, 103134.
5. Gržinić, T.; González, E.B. Methods for Automatic Malware Analysis and Classification: A Survey. Int. J. Inf. Comput. Secur. 2022, 17, 179–203.
6. Aslan, Ö.A.; Samet, R. A Comprehensive Review on Malware Detection Approaches. IEEE Access 2020, 8, 6249–6271.
7. Muzaffar, A.; Hassen, H.R.; Lones, M.A.; Zantout, H. An In-Depth Review of Machine Learning Based Android Malware Detection. Comput. Secur. 2022, 121, 102833.
8. Firdausi, I.; Erwin, A.; Nugroho, A.S. Analysis of Machine Learning Techniques Used in Behavior-Based Malware Detection. In Proceedings of the 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, Jakarta, Indonesia, 2–3 December 2010; pp. 201–203.
9. Fuyong, Z.; Tiezhu, Z. Malware Detection and Classification Based on N-Grams Attribute Similarity. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 1, pp. 793–796.
10. Taheri, L.; Kadir, A.F.A.; Lashkari, A.H. Extensible Android Malware Detection and Family Classification Using Network-Flows and API-Calls. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8.
11. Mu, T.; Chen, H.; Du, J.; Xu, A. An Android Malware Detection Method Using Deep Learning Based on Api Calls. In Proceedings of the 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 11–13 October 2019; pp. 2001–2004.
12. Tran, T.K.; Sato, H.; Kubo, M. Image-Based Unknown Malware Classification with Few-Shot Learning Models. In Proceedings of the 2019 Seventh International Symposium on Computing

and Networking Workshops (CANDARW), Nagasaki, Japan, 26–29 November 2019; pp. 401–407.

13. Makandar, A.; Patrot, A. Malware Class Recognition Using Image Processing Techniques. In Proceedings of the 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), Pune, India, 24–26 February 2017; pp. 76–80.
14. Tran, T.K.; Sato, H. NLP-Based Approaches for Malware Classification from API Sequences. In Proceedings of the 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES), Hanoi, Vietnam, 15–17 November 2017; pp. 101–105.
15. Nagano, Y.; Uda, R. Static Analysis with Paragraph Vector for Malware Detection. In Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, Beppu, Japan, 5–7 January 2017; pp. 1–7.
16. Schofield, M.; Alicioglu, G.; Binaco, R.; Turner, P.; Thatcher, C.; Lam, A.; Sun, B. Convolutional Neural Network for Malware Classification Based on API Call Sequence. In Proceedings of the 8th International Conference on Artificial Intelligence and Applications (AIAP 2021), Zurich, Switzerland, 23–24 January 2021; pp. 23–24.
17. Ravi, C.; Manoharan, R. Malware Detection Using Windows Api Sequence and Machine Learning. Int. J. Comput. Appl. 2012, 43, 12–16.
18. Nakazato, J.; Song, J.; Eto, M.; Inoue, D.; Nakao, K. A Novel Malware Clustering Method Using Frequency of Function Call Traces in Parallel Threads. IEICE Trans. Inf. Syst. 2011, 94, 2150–2158.
19. Kolosnjaji, B.; Zarras, A.; Webster, G.; Eckert, C. Deep Learning for Classification of Malware System Call Sequences. In Proceedings of the AI 2016: Advances in Artificial Intelligence: 29th Australasian Joint Conference, Hobart, TAS, Australia, 5–8 December 2016; Proceedings 29. Springer: Berlin/Heidelberg, Germany, 2016; pp. 137–149.
20. Li, C.; Zheng, J. API Call-Based Malware Classification Using Recurrent Neural Networks. J. Cyber Secur. Mobil. 2021, 10, 617–640.
21. Li, C.; Lv, Q.; Li, N.; Wang, Y.; Sun, D.; Qiao, Y. A Novel Deep Framework for Dynamic Malware Detection Based on API Sequence Intrinsic Features. Comput. Secur. 2022, 116, 102686.
22. Li, C.; Cheng, Z.; Zhu, H.; Wang, L.; Lv, Q.; Wang, Y.; Li, N.; Sun, D. DMalNet: Dynamic Malware Analysis Based on API Feature Engineering and Graph Learning. Comput. Secur. 2022, 122, 102872.
23. Daeef, A.Y.; Al-Naji, A.; Chahl, J. Features Engineering for Malware Family Classification Based API Call. Computers 2022, 11, 160.

24. Deore, M.; Kulkarni, U. Mdfrcnn: Malware Detection Using Faster Region Proposals Convolution Neural Network. Int. J. Interact. Multimedia Artif. Intell. 2022, 7, 146–162.
25. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. arXiv 2014, arXiv:1406.1078.
26. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A Tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv 2019, arXiv:1909.09586.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. Adv. Neural Inf. Process. Syst. 2017, 30.
28. Zeyer, A.; Bahar, P.; Irie, K.; Schlüter, R.; Ney, H. A Comparison of Transformer and Lstm Encoder Decoder Models for Asr. In Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019; pp. 8–15.
29. Rahali, A.; Akhloufi, M.A. MalBERT: Malware Detection Using Bidirectional Encoder Representations from Transformers. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 3226–3231.
30. Yang, X.; Yang, D.; Li, Y. A Hybrid Attention Network for Malware Detection Based on Multi-Feature Aligned and Fusion. Electronics 2023, 12, 713.
31. Ma, Q.; Yu, L.; Tian, S.; Chen, E.; Ng, W.W. Global-Local Mutual Attention Model for Text Classification. IEEE/ACM Trans. Audio Speech Lang. Process. 2019, 27, 2127–2139.
32. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.
33. DeVries, T.; Taylor, G.W. Improved Regularization of Convolutional Neural Networks with Cutout. arXiv 2017, arXiv:1708.04552.
34. Sung, K.-K.; Poggio, T. Example-Based Learning for View-Based Human Face Detection. IEEE Trans. Pattern Anal. Mach. Intell. 1998, 20, 39–51.
35. Shrivastava, A.; Gupta, A.; Girshick, R. Training Region-Based Object Detectors with Online Hard Example Mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.
36. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An Advanced Object Detection Network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
37. Hwang, J.; Kim, J.; Lee, S.; Kim, K. Two-Stage Ransomware Detection Using Dynamic Analysis and Machine Learning Techniques. Wirel. Pers. Commun. 2020, 112, 2597–2609.

38. Baek, S.; Jeon, J.; Jeong, B.; Jeong, Y.-S. Two-Stage Hybrid Malware Detection Using Deep Learning. Hum. Centric Comput. Inf. Sci. 2021, 11, 10–22967.