Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Machine Learning for Malware Detection and its State-of-the-Art and Future Challenges

Charmy khapra

Guru Gobind Singh Indraprastha University, Charmykhapra@gmail.com

Malware has become a significant cyber threat which constantly changes to smart devices, target systems, and large-scale networks with proliferation of latest technologies. Malware detection has definitely been a challenge and major concern due to shortcomings in analysis type, performance accuracy, and malware detection methods that cannot detect unexpected attacks. This study seeks to perform a thorough review to provide a "taxonomy of machine learning (ML) approaches to detect malware. This study investigates ML and malware in context of cybersecurity, such as classification of ML model and taxonomy of malware detection in various categories.
In addition, the taxonomy could evaluate the most recent models for machine learning. This study has also determined the concerns and challenges related to malware detection as well as remedies. Finally, to deal with relevant concerns to encourage researchers in future studies, this study fills the gap by offering complete state-of-the-art of malware detection. This study introduces a taxonomy of feature representation and feature extraction along with malware detection approaches and associates each approach with most widely used types of data. The approach for feature extraction is introduced as per the techniques used rather than analysis method.

Keywords: machine learning, malware detection, cybersecurity, taxonomy, feature extraction, feature representation.

1. Introduction

The COVID-19 outbreak has significantly increased cyber-attacks across the world, which has also brought a phenomenon called "cyber pandemic" which was predicted to cross US$10.5 trillion by 2025 in terms of cost of annual damage (Fichtenkamm et al, 2022; Morgan, 2022). Meanwhile, the pandemic has imposed a new "work-from-home" model which has increased threat exposure to most organizations significantly (Julisch et al, 2020). Before the pandemic, around 20% of attackers used unexpected attack or malware practices, as many of these having "machine learning (ML)" models to stay hidden and blend in the environment. During the pandemic, there is 35% rise in this proportion (Nabe, 2022). These trends, together, indicate negative path and impact a lot of businesses across all industries and can advance at unexpected rate.

With exponential rise in computing power, data and technology, more and more smart tools should be used to deal with rising problems. As humans are unable to manage the rising

problems on themselves, it is not possible to avoid the dependence on AI. Market for AI is expected to rise from US$3.92 billion in the year 2017 to US$34.81 billion by the year 2025 in cybersecurity (Loaiza et al, 2019). In addition, the "Capgemini Research Institute" has conducted a survey and observed that 69% of companies feel that AI is required to deal with cyberattacks (Belani, 2021). AI is gaining a lot of attention from private and public sectors. However, its power will definitely fall into cybercriminals' hands, resulting in next-gen AI-based malware.

This unexpected upsurge of AI makes it very important for experts to detect cybercriminal activities and malware smoothly. Irrespective of a lot of advancements in AI-based models for malware detection, this progress is insufficient and a lot of efforts are needed to outperform cybercriminals. There are different kinds of malware like worms, viruses, trojans, spyware, and ransomware. They are always aimed to compromise data and systems or demand ransom from the victim.

2. Literature Review

Initially, "Adversarial Machine Learning (AML)" was introduced when researchers found some blind spots in image classifiers in the field of computer vision which were used by such adversarial samples to betray the model. Even though it has been researched well in the field of computer vision, AML has significant impact in mobile and wireless communication systems. The application of machine learning classifiers have showed different benefits to mobile and wireless networks to detect malware and anomalies in network traffic. However, those detectors can also be evaded or tricked by well-designed samples by attackers, raising security issues for applications. Hence, it has become vital to detect those samples to protect the network. Liu et al. (2021) presented a comprehensive and systematic review on AML to mobile and wireless systems from physical to application and network layers. The researchers have reviewed modern approaches to detect and generate adversarial samples. Adversarial models are used to generate samples like "Generative Adversarial Networks (GANS)" and "Fast Gradient Sign Method (FGSM)" techniques. Adversarial models can be used to detect samples which act as ML classifiers or classifiers reinforced with knowledge on detecting those samples. They discussed and highlighted challenges and open issues for those approaches and opportunities which are interested to developers and researchers in AI-based mobile and wireless networking.

Saad et al. (2019) argued that ML techniques are not prepared for malware detection directly. With existing trend in rise in malware and attacks, dynamic malware analysis is required for antimalware detection. The researchers have presented a complete review of ML for malware detection. They discussed how "malware detection in the wild" has different challenges for existing ML techniques. Three important problems have been defined that can affect malware detectors' success due to "machine learning in the wild." They also explored potential solutions to those challenges and presented the need for next-gen malware detection. Finally, potential research direction have been discussed for malware detection in machine learning.

Every year, malware is one of the significant cybersecurity concerns since complexity of malware is changing rapidly with innovation. Hence, malware attacks have affected the lives

through different ways and mediums. Hence, an ML model is among the vital solutions to secure systems for malware detection when it comes to the ability of ML models to stay ahead with evolution. Gorment et al. (2021) reviewed the most updated works from 2017 to 2021 on using ML models for malware detection like "Decision Tree, K-Means, Naïve Bayes, Meta-Heuristic, Neuro-fuzzy, Gaussian, Bayesian, K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and n-Grams" with a systematic literature review. This study describes each ML model, showing malware detection performance for each model, and presented limitations and challenges of model.

Alqahtani (2021) discussed several ML and AI techniques for malware detection. Traditional ML techniques like KNN, SVM, and decision tree as well as advanced techniques like CNN and ANN are explored. They achieved 99% of highest accuracy followed by "98.422%, 97.3%, and 96%" where "package-level API calls" have been implemented as feature, followed by modern classification approach. In addition, the researchers have discussed dataset details for testing with malware detection. "DREBIN" is one of those datasets with largest number of samples having "5560 Malware samples" and "123453 Benign samples". Finally, the researchers have discussed open challenges and highlighted future directions which encourage a researcher to solve these challenges and field of research with future directions.

Gibert et al. (2020) provided an in-depth and systematic view of ML techniques for detecting malware and, especially deep learning. This study provided a complete overview of features and approaches in a traditional workflow of machine learning for malware classification and detection. They also explored limitations and challenges of traditional ML techniques, and analysed recent developments and trends while focusing on DL models. In addition, they presented unsolved challenges and research issues of modern techniques as well as new directions of research. This study helps researchers to understand the field of malware detection and of new directions of research and developments explored by scientific community to deal with the issue.

2.1 Research Gap

Malware detection was completely based on calculation of constant signatures and byte sequences of suspected file to the signature of known database or malware. As recent "polymorphic" malware comes out, "signature-based detection" has been superseded and less effective by heuristic-based, next-generation, and behavioral-based methods depending on ML models (Ray, 2021). All the leading malware detection solutions are underpinned by ML models, also known as Endpoint Detection and Response (EDR)" (Mellen, 2022). This study fills the research gap by providing and analysing recent efforts in more effective ways to use ML for malware detection. It will help researchers to come up with knowledge in the field of malware detection and new research directions and developments.

2.2 Research Objectives

- To discuss various ML models and data types for malware detection
- To investigate the taxonomy of Representation Methods and Feature Extraction

3. Research Methodology

This study has collected secondary data as per the SLR standards proposed by Moher et al. (2009) and Keele (2007). The researchers have built an automatic strategy as per the research objectives. The search methods and syntax from various online libraries have been used to search the database and address the string of queries. Secondary data has been collected from various digital libraries and databases like Mendeley, IEEE Explore, Scopus, ACM Digital Library, SpringerLink, Web of Science, and Science Direct. These digital libraries were used to identify the relevant studies during the search stage. After fulfilling the inclusion and exclusion criteria, all the articles were sorted for further research. Most up-to-date and recent papers have been selected.

3.1 Inclusion and Exclusion Criteria

Inclusion and exclusion criteria have been developed as per the research objectives to narrow down the relevant studies for this research. There are three steps in the selection process. First of all, only relevant studies have been selected and then researchers have read the abstracts and titles of all papers.

The inclusion criteria include the following studies –

- Studies which have applied or described ML models for malware detection and relevant processes
- Studies were summarized from a review paper and each publication had got its treatment
- Studies should be recent

Here is the exclusion criteria for the study –

- The study discusses ML models but not used for malware detection
- Studies similar to existing paper or duplicate paper
- Articles which were not published in English language

3.2 Data Extraction

The research objectives have been fulfilled with detailed analysis of available data which is collected. As per the selected studies, here are the data extracted for this study –

- Name of publication
- Country
- Discipline
- Type of machine learning approach for malware detection
- Models, ideas, and algorithms which are important
- Identifying ML model with specific classification analysis and approach
- Tools used for malware detection

The review processes are based on the SLR standards derived by Kitchenham [9-10]: search strategy, inclusion and exclusion criteria, quality evaluation, data extraction, and data analysis.

4. Data Analysis

4.1 ML models and data types for malware detection

Malware analysis and data type have a significant and rising effect on the process of malware detection to evaluate the classification of file investigated and, hence, impacts the overall accuracy of the models. Various data types have been extracted with dynamic, static, and hybrid analysis like "Opcode, Byte code, file data, API calls, and registry data, to acknowledge and understand the important function and purpose of files tested and they are classified as benign or malware files. Hence, here are some of the analysis methods used in recent studies along with the data types extracted in recent studies –

- Static analysis – This approach of malware analysis has been used widely to discover the source code running practicable files to gather unique signature to represent the file to be investigated. Various static data can be gathered through static analysis like compression ratio, derived data, and PE-header data (Naz & Singh, 2019; Zelinka & Amer, 2019; Denzer et al, 2019; Ling et al, 2019; Kumar et al, 2019). In addition, static tools for analysis like modules developed with Python and "IDA-pro disassembler" are widely used to gather API calls and static opcode (Euh et al, 2020). Encryption and packing techniques can influence the potential execution paths.

- Dynamic Analysis – This kind of analysis approach is performed by various researchers to gather several data types to differentiate between benign and malware files by running the files that are executable in virtual machines (VM), isolated environments, or emulators to track the behavior of executable file during run-time and gather the needed dynamic data (Choudhary & Vidyarthi, 2015).

- Hybrid analysis – Data extracted with dynamic and static analysis have been combined to reduce the issues of both approaches and achieve higher accuracy. Both static and dynamic data are collected by various tools like IDA pro disassembler, Cuckoo sandbox, and OlleyDbg. Then, the sets of hybrid features are created as per various data types like opcode, string, API calls, etc. (Ndibanje et al, 2019; Zhong & Gu, 2019; Huang et al, 2021).

Malware detection refers to the system that should be implemented to identify and discover malicious activities of files being investigated. Hence, various malware detection approaches have been improved every year without any approach offering 100% success with all types and families of malware in every situation. Hence, signatures and behaviors are two important characteristics to detect malicious software with three approaches –

4.1.1 Signature-based malware detection

Various studies have been conducted to improve classification and malware detection models

by depending on unique signature which has been dynamically or statically extracted and stored to compare the same with signature of investigated file which is collected. These signatures consist of a range of API calls, byte-code series, opcodes, and entropy quantity. Wael et al. (2017; 2018) have generated static "string-based signatures" to detect "VBasic malicious program" by defining the strings obtained with frequency vectors (Fuyong & Tiezhu, 2017).

4.1.2 Behavior based detection

Once executable files are monitored in an individual environment and exhibited behaviors are collected, features extraction has been developed to collect sensitive features. This way, the malicious behaviors can easily be classified by the developed model along with any behavior which is supposed to be similar related to false positive signs.

4.1.3 Heuristic-based approach

A heuristic approach is used in different studies when generic rules are generated to investigate the data extracted with static and dynamic analysis for supporting the proposed model for identifying malicious intent. The generated rules can be automatically developed with ML models or YARA tool which is manually based on knowledge and experience of analysts (Mosli et al, 2016; Jerlin & Marimuthu, 2018; Belaoued et al, 2019).

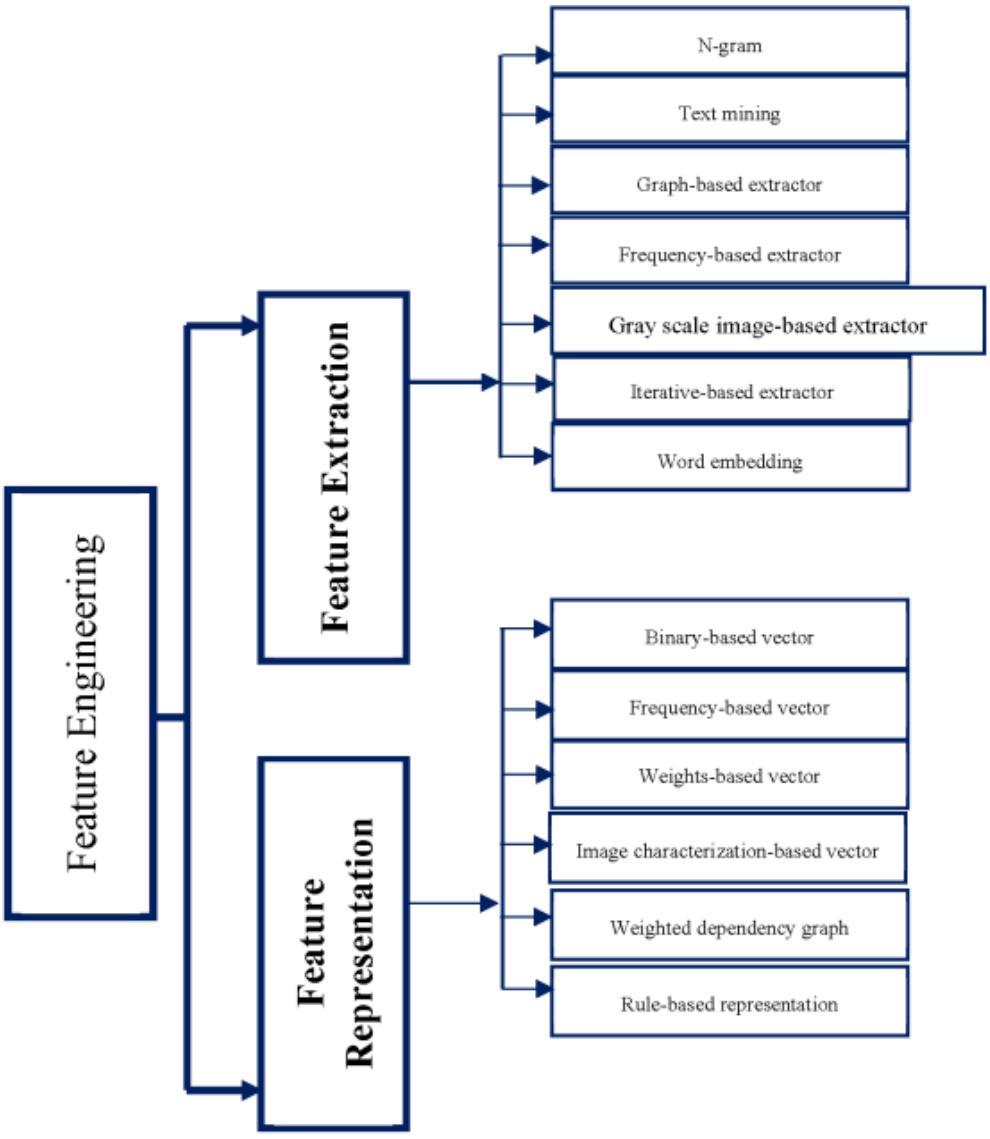
4.2 Taxonomy of Representation Methods and Feature Extraction

This section consists of taxonomy of feature representation and extraction approaches. In addition, the taxonomy of representing novel feature is identified to clear the borders among data extraction, collection, and representation stages. Figure 1 illustrates the taxonomy of feature extraction and representation.

4.2.1 Feature Extraction

Feature engineering includes selection, extraction, and representation of features. It is an important stage in the process of malware detection and classification as it has a drastic impact on the performance of classification model (Ali et al, 2020). After all, the procedure of feature engineering arranges features for subset of data which is easy to understand for machines (Ranveer & Hiray, 2015). In addition, the computational overhead can be reduced by reducing dataset for processing (Shabtai et al, 2009).

Figure 1 – Taxonomy of Feature Representation and Feature Extraction



Source – Aboaoja et al. (2022)

- N-Gram – This technique has been widely used by various studies for feature extraction (Ahmed et al, 2020; Liu et al, 2017; Shijo et al, 2015; Ding et al, 2018; More & Gaikwad, 2016). Subsets are built from the original text set with "length of n". As per the application, there are different types of data included in the string. It can include words or letters. N-grams divide the text string in "fixed-length substrings". The N-gram can be used to measure the accuracy in similarity across terms (Zhao et al, 2019). The "n-gram technique" is used to cause "high dimensionality feature space" because of a lot of features generated which result in more time to be used (Kakisim et al, 2019).

- Text mining – This technique is taken from the field of retrieving data. Term weighting and indexing include two important categories of techniques for text-mining. Term indexing is assigned to each term and term weighting refers to calculation of specific weight. Choudhary & Vidyarthi (2015) assigned weights in analysis report for each text with a technique to gain information. They assigned those weights to the text representing the operations performed and their locations as per occurrences of such texts in benign and malware categories. Additionally, the "Term Frequency-Inverse Document Frequency (TF-IDF)" has been used as a weighting approach to gather network traffic and API cells by determining each feature in both benign and malware categories (Belaoued et al, 2019).

- Graph-based Extractor – To extract the most vital features only, raw data like opcodes and API calls are formed in graphical format representing the arrival of each feature in the sample. Hence, constant sub-graphs among all sample family or class are known to be important feature for extraction. Those sub-graphs include opcodes and API calls and they are related to control flow or dependency (Allen, 1970).

- Frequency-based extractor – In the raw data, each feature can be redundant or discriminative. The features are important as per how they take place in each category. As long as feature comes out frequently in a class and doesn't appear in other categories, there is a significant rise in that feature (Kang & Won, 2020; Zhang et al, 2019).

- Word Embedding – This method is capable to predict distribution of each feature or word. Some of the popular embedding techniques like Word2Vec can be developed with Skip-Gram and Bag-of-Words (CBOW)" used the context before and after target word for prediction of output (Amer & Zelinka, 2020).

- Iterative-based Extractor – Since final objective of each classification and malware detection model is achieving ideal accuracy when test data is classified, this approach is based on features to improve the detection accuracy.

- Gray-scale extractor – Each single pixel or 8-bit unit in the image of binary can be painted as per the degree of color in that pixel. This way, each malware binary generates gray images and there are different features that can be extracted with the generated images like intensity, texture, and wavelet (Nataraj et al, 2011).

4.2.2 Feature Representation

it is a very important step after feature extraction. It includes how to transfer the characteristics extracted which are easy to understand for machines and models. It refers to transfer of characteristics which are extracted to be comprehensible for machines and models with various types of vectors. This way, proposed models can learn various class behaviors (Shabtai et al, 2009).

- Binary-based vector – This representation method gathers features extracted from the point of view of either true or false. If feature examined exists in document, it is considered true. If feature examined doesn't exist, it is false (Sihwail et al, 2019).

- Frequency-based vector – These days, the malware production is simple with different online tools and code libraries in malicious software that can alter and reuse the current malicious codes. Hence, frequency of each feature extracted in each sample of malware like

opcodes and API calls can help identify similar malicious behaviors from various models (Galal et al, 2016).

- Weight-based vector – Extracted feature distribution is among the major differences among malware and legitimate categories and among malware families. The features associated with malicious behavior should be distributed heavily in samples and are not in valid samples (Banin & Dyrkolbotn, 2018).

- Image characterization vector – To generate vectors based on image characteristics and malware files are represented to describe the malware images, there are different models like "Homogeneous Texture Descriptor (HTD), Color Layout Descriptor (CLD), etc." are used to obtain the vector from images generated. The generated vector can define the created images of benign and malware files should have different values as they represent several image aspects like intensity and texture (Kang & Won, 2020).

- Weighted Dependency – Since malware classification and detection models have been developed as per specific sequences or features, they are vulnerable to unreadable reordering and insertion techniques. More complex details like feature dependencies should be learned and included by models developed (Ding et al, 2018).

- Rule-based representation – Most of the mechanisms for voting decision-making have been used widely in earlier studies to find out if rule-based behavior of text file is more aligned with malware or benign behavior. However, these models can detect only malicious activities defined in generated rules (Belaoued et al, 2019).

5. Discussion and Conclusion

As per the recent malware classification and detection models, this study has discussed various techniques and approaches, their benefits and usefulness. There are different techniques which were once used to protect IP of software vendors are now used by attackers to write malicious codes to transfer the malware to various forms which are harder to detect and analyze. Some of those techniques are intrusion reordering, inserting dead code, and instruction substitution to update malware characteristics apart from achieving the same goals. While performing specific operations, the evasive malware can recognize whether they are running in a real or controlled environment.

There are different solutions which have opened the path to develop trusted malware classification and detection models. Various methods have been considered to detect evasive malware like generating API-based signatures for evasive malware, detecting evasion with various execution settings, and some of the popular evasion techniques for evasive malware detection. This study has explored the trends related to malware detection and analysis. This study has also presented a novel taxonomy of feature representation methods.

References

1. Fichtenkamm, M. Burch, G., & Burch, J. (2022). Cybersecurity in a COVID-19 World: Insights on How Decisions Are Made. libraryguides.vu.edu.au. https://www.isaca.org/resources/isacajournal/issues/2022/volume-2/cybersecurity-in-a-covid-

19-world.
2. Morgan, S. (2022). Cybercrime to Cost the World $10.5 Trillion Annually by 2025. Cybersecurity Ventures. Available at https://cybersecurityventures.com/hackerpocalypse-cybercrimereport-2016/.
3. Julisch, K. Widmer, F. & Grampp, M. (2020). Cybercrime – the risks of working from home. Deloitte. Retrieved from https://www2.deloitte.com/ch/en/pages/risk/articles/covid-19-cyber-crime-working-from-home.html.
4. Nabe, C. (2022). Impact of COVID-19 on Cybersecurity. Deloitte.com. Available at https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html
5. Loaiza, F. L., Birdwell, J. D., Kennedy, G. L., & Visser, D. (2019). Utility of artificial intelligence and machine learning in cybersecurity. Institute for Defense Analyses..
6. Belani, G. (2021). The use of artificial intelligence in cybersecurity: A review. IEEE Computer Society. Available at https://www.computer.org/publications/tech-news/trends/the-use-of-artificial-intelligence-incybersecurity.
7. Liu, J., Nogueira, M., Fernandes, J., & Kantarci, B. (2021). Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems. IEEE Communications Surveys & Tutorials, 24(1), 123-159.
8. Saad, S., Briguglio, W., & Elmiligi, H. (2019). The curious case of machine learning in malware detection. arXiv preprint arXiv:1905.07573.
9. Gorment, N. Z., Selamat, A., & Krejcar, O. (2021). A recent research on malware detection using machine learning algorithm: Current challenges and future works. In Advances in Visual Informatics: 7th International Visual Informatics Conference, IVIC 2021, Kajang, Malaysia, November 23–25, 2021, Proceedings 7 (pp. 469-481). Springer International Publishing.
10. Alqahtani, M. A. (2021). Machine learning techniques for malware detection with challenges and future directions. International Journal of Communication Networks and Information Security, 13(2), 258-270.
11. Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. Journal of Network and Computer Applications, 153, 102526.
12. Ray, A. (2021). Cybersecurity for connected medical devices. Academic Press.
13. Mellen, A. (2022). The Forrester WaveTM: Endpoint Detection And Response Providers, Q2 2022. Forrester. Retrieved from https://www.forrester.com/report/the-forrester-wave-tm-endpoint-detectionand-response-providers-q2-2022/RES176332?reference=twitter&utm_source=twitter&utm_medium=ppc&utm_campaign=msbg_cx_cert
14. Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering (Vol. 5). Technical report, ver. 2.3 ebse technical report. ebse.
15. Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & PRISMA Group*, T. (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. Annals of internal medicine, 151(4), 264-269.
16. Naz, S., & Singh, D. K. (2019, July). Review of machine learning methods for windows malware detection. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.
17. Zelinka, I., & Amer, E. (2019, December). An ensemble-based malware detection model using minimum feature set. In Mendel (Vol. 25, No. 2, pp. 1-10).
18. Denzer, T., Shalaginov, A., & Dyrkolbotn, G. O. (2019). Intelligent windows malware type detection based on multiple sources of dynamic characteristics. Nis. J, 12(20).
19. Ling, Y. T., Sani, N. F. M., Abdullah, M. T., & Hamid, N. A. W. A. (2019). Nonnegative matrix factorization and metamorphic malware detection. Journal of Computer Virology and Hacking Techniques, 15, 195-208.

20. Kumar, A., Kuppusamy, K. S., & Aghila, G. (2019). A learning model to detect maliciousness of portable executable using integrated feature set. Journal of King Saud University-Computer and Information Sciences, 31(2), 252-265.
21. Euh, S., Lee, H., Kim, D., & Hwang, D. (2020). Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems. IEEE Access, 8, 76796-76808.
22. Ndibanje, B., Kim, K. H., Kang, Y. J., Kim, H. H., Kim, T. Y., & Lee, H. J. (2019). Cross-method-based analysis and classification of malicious behavior by api calls extraction. Applied Sciences, 9(2), 239.
23. Zhong, W., & Gu, F. (2019). A multi-level deep learning system for malware detection. Expert Systems with Applications, 133, 151-162.
24. Huang, X., Ma, L., Yang, W., & Zhong, Y. (2021). A method for windows malware detection based on deep learning. Journal of Signal Processing Systems, 93, 265-273.
25. Wael, D., Sayed, S. G., & AbdelBaki, N. (2018). Enhanced approach to detect malicious vbscript files based on data mining techniques. Procedia Computer Science, 141, 552-558.
26. Wael, D., Shosha, A., & Sayed, S. G. (2017, November). Malicious vbscript detection algorithm based on data-mining techniques. In 2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT) (pp. 112-116). IEEE.
27. Fuyong, Z., & Tiezhu, Z. (2017, July). Malware detection and classification based on n-grams attribute similarity. In 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC) (Vol. 1, pp. 793-796). IEEE.
28. Mosli, R., Li, R., Yuan, B., & Pan, Y. (2016, May). Automated malware detection using artifacts in forensic memory images. In 2016 IEEE Symposium on Technologies for Homeland Security (HST) (pp. 1-6). IEEE.
29. Jerlin, M. A., & Marimuthu, K. (2018). A new malware detection system using machine learning techniques for API call sequences. Journal of Applied Security Research, 13(1), 45-62.
30. Belaoued, M., Boukellal, A., Koalal, M. A., Derhab, A., Mazouzi, S., & Khan, F. A. (2019). Combined dynamic multi-feature and rule-based behavior for accurate malware detection. International Journal of Distributed Sensor Networks, 15(11), 1550147719889907.
31. Ali, M., Shiaeles, S., Bendiab, G., & Ghita, B. (2020). MALGRA: Machine learning and N-gram malware feature extraction and detection system. Electronics, 9(11), 1777.
32. Ranveer, S., & Hiray, S. (2015). Comparative analysis of feature extraction methods of malware detection. International Journal of Computer Applications, 120(5).
33. Shabtai, A., Moskovitch, R., Elovici, Y., & Glezer, C. (2009). Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. information security technical report, 14(1), 16-29.
34. Ahmed, Y. A., Koçer, B., Huda, S., Al-rimy, B. A. S., & Hassan, M. M. (2020). A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection. Journal of Network and Computer Applications, 167, 102753.
35. Liu, L., Wang, B. S., Yu, B., & Zhong, Q. X. (2017). Automatic malware classification and new malware detection using machine learning. Frontiers of Information Technology & Electronic Engineering, 18(9), 1336-1347.
36. Shijo, P. V., & Salim, A. J. P. C. S. (2015). Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 46, 804-811.
37. Ding, Y., Xia, X., Chen, S., & Li, Y. (2018). A malware detection method based on family behavior graph. Computers & Security, 73, 73-86.
38. More, S. S., & Gaikwad, P. P. (2016). Trust-based voting method for efficient malware detection. Procedia Computer Science, 79, 657-667.
39. Zhao, Y., Bo, B., Feng, Y., Xu, C., & Yu, B. (2019). A feature extraction method of hybrid gram

for malicious behavior based on machine learning. Security and Communication Networks, 2019(1), 2674684.
40. Kakisim, A. G., Nar, M., & Sogukpinar, I. (2020). Metamorphic malware identification using engine-specific patterns based on co-opcode graphs. Computer Standards & Interfaces, 71, 103443.
41. Choudhary, S. P., & Vidyarthi, M. D. (2015). A simple method for detection of metamorphic malware using dynamic analysis and text mining. Procedia Computer Science, 54, 265-270.
42. Belaoued, M., Boukellal, A., Koalal, M. A., Derhab, A., Mazouzi, S., & Khan, F. A. (2019). Combined dynamic multi-feature and rule-based behavior for accurate malware detection. International Journal of Distributed Sensor Networks, 15(11), 1550147719889907.
43. Aboaoja, F. A., Zainal, A., Ghaleb, F. A., Al-Rimy, B. A. S., Eisa, T. A. E., & Elnour, A. A. H. (2022). Malware detection issues, challenges, and future directions: A survey. Applied Sciences, 12(17), 8482.
44. Allen, F. E. (1970). Control flow analysis. ACM Sigplan Notices, 5(7), 1-19.
45. Kang, J., & Won, Y. (2020). A study on variant malware detection techniques using static and dynamic features. Journal of Information Processing Systems, 16(4), 882-895.
46. Zhang, J., Qin, Z., Yin, H., Ou, L., & Zhang, K. (2019). A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding. Computers & Security, 84, 376-392.
47. Amer, E., & Zelinka, I. (2020). A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence. Computers & Security, 92, 101760.
48. Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In Proceedings of the 8th international symposium on visualization for cyber security (pp. 1-7).
49. Shabtai, A., Moskovitch, R., Elovici, Y., & Glezer, C. (2009). Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. information security technical report, 14(1), 16-29.
50. Sihwail, R., Omar, K., Zainol Ariffin, K. A., & Al Afghani, S. (2019). Malware detection approach based on artifacts in memory image and dynamic analysis. Applied Sciences, 9(18), 3680.
51. Galal, H. S., Mahdy, Y. B., & Atiea, M. A. (2016). Behavior-based features model for malware detection. Journal of Computer Virology and Hacking Techniques, 12, 59-67.
52. Banin, S., & Dyrkolbotn, G. O. (2018). Multinomial malware classification via low-level features. Digital Investigation, 26, S107-S117.
53. Kang, J., & Won, Y. (2020). A study on variant malware detection techniques using static and dynamic features. Journal of Information Processing Systems, 16(4), 882-895.
54. Ding, Y., Xia, X., Chen, S., & Li, Y. (2018). A malware detection method based on family behavior graph. Computers & Security, 73, 73-86.
55. Belaoued, M., Boukellal, A., Koalal, M. A., Derhab, A., Mazouzi, S., & Khan, F. A. (2019). Combined dynamic multi-feature and rule-based behavior for accurate malware detection. International Journal of Distributed Sensor Networks, 15(11), 1550147719889907.