Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 9s (2025)

A Comprehensive Survey on Polymorphic Malware Analysis: Challenges, Techniques, and Future Directions

Dr. Madhavi Satish Avhankar[1], Dr. Janardan Pawar[2], Dr. Vijaya Kumbhar[3]

[1,2]Computer Science Indira College of Commerce and Science Pune, India,

[3]School of Computer Studies, Sri Balaji University, Pune, India

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract: Since the beginning of computing, malicious software has changed dramatically, becoming more complex and elusive. The increase in ransomware attacks has brought attention to the serious risks that malware poses, affecting not only individuals but also organizations, governments, and vital infrastructure like transportation networks and hospitals. Mitigating these dangers requires early identification of harmful behaviour, yet detecting new and unknown malware is still quite difficult. Static and dynamic analysis are the two main types of malware analysis approaches. Dynamic analysis watches how a file behaves in a controlled setting, whereas static analysis looks at a file without running it. Static analysis is less successful since malware writers use evasion strategies including dynamic code loading, encryption, and code obfuscation to evade detection. Conversely, dynamic analysis improves detection capabilities and provides deeper insights into malware behaviour while offering resilience against such evasion tactics. Notwithstanding these benefits, no one method is infallible, and current technologies are not always able to adequately capture the intricacies of polymorphic malware.

The methods currently used to analyse polymorphic malware are thoroughly reviewed in this survey, with an emphasis on their advantages, disadvantages, and room for development. This study intends to aid in the creation of more resilient and flexible malware detection systems by assessing the efficacy of different analytical methodologies.

Keywords: polymorphic malicious software, static analysis, dynamic analysis.

1. Introduction

Attacks using harmful software, or malware, are becoming more frequent at a startling rate. Malware is growing more and more common, impacting both large organizations and individual users. It poses a serious security risk to sensitive data and priceless computer resources [1]. A startling 560,000 new cases of malware are found every day, and there are currently more than 1 billion malware programs available. Nearly four businesses are attacked by ransomware every day, and 58% of all malware connected to computers is Trojan horses. Indeed, there is reason for alarm regarding the express spectrum of malware. Every new dangerous application that is discovered by antimalware groups is added to their databases, which are constantly growing. Every day, hundreds of millions of files on computers and websites are infected with malware, which is mostly caused by recurring threats that spread like infectious diseases. According to recent estimates, about 17 million new malware occurrences are discovered each month. [1]

According to studies from the AV-TEST organization, more than 450,000 newly discovered dangerous applications (malware) and potentially unwanted apps (PUA) are registered every day. [2] Over time, cyberattacks have evolved into more intricate and targeted operations. The goals of these focused attacks are to compromise networks, steal confidential information, interfere with normal business operations, and steal private financial data. Well-known malware detection methods are quite effective at recognizing recognized signatures. But a new threat has surfaced in the shape of polymorphic malware, which uses strategies to continuously change its program characteristics (signatures) in order to avoid detection. First appearing in 1990, polymorphic malware can take many different forms and create malicious code by using polymorphic engines. These malware variants are significantly more evasive and difficult for antivirus software to identify.

Unlike regular harmful software that antivirus software can detect, polymorphic malware has a different effect on software programs. The first to come to light was capable of self-alteration and self-decryption, but it also created some hazardous dangers that were not detectable with signature-based methods. Additionally, malware authors use a variety of strategies to make an infinite number of harmful attempts every day. The application of code insertion and other strategies, such as obfuscation, has enabled these developments. These writers use polymorphic toolkits, such as mutating engines and polymorphic packs, also referred to as polymorphism engines, to transform non-obfuscation malware into polymorphic [3].

Because polymorphic malware is largely undetectable, traditional protections that rely on signatures are helpless against it. The current range of detection efficacy for polymorphic malware is 68.75% to 81.25%, highlighting the urgent need for novel detection techniques. Expanding beyond signature-based detection is the only way to enhance the detection of polymorphic malware. In order to accurately identify polymorphic malware through advanced analysis, it is necessary to extract key properties and characteristics.

The following are the research's goal:

1. To conduct a thorough examination of polymorphic malware analysis methods and resources

2. To learn how cyber security risks operate.

3. To thoroughly understand every pertinent aspect of polymorphic malicious software and do a thorough deconstruction of it.

4. To fully comprehend how polymorphic malware analysis works and behaves. Malware researchers have traditionally focused on one-dimensional elements like source/execution code analysis (static) or behavior (dynamic).

To support existing classification and detection techniques, more research is necessary. To authentically depict the inner workings of the malware, this research should place a high priority on the thorough extraction of all relevant features and their careful dissection. [3] The rest of the paper is organized as follows: Section II provides examples of the related area of polymorphic malware analysis. A discussion and highlights of the study are included in Section III, and the report is concluded in Section IV.

2. Literature Review

Compared to regular malicious software, which antivirus programs can identify, polymorphic malware has a more negative effect on software applications. By decrypting itself, the first version of polymorphic malware demonstrated its versatility and introduced a few dangerous risks that were unable to detect by signature-based security techniques. Using methods like encryption, data appending or pre-pending, and altering decryption procedures with every infection, polymorphic malware alters its fundamental structure as long as the encryption key does. As a result, once polymorphic malware enters computer systems, new code is created, making it very difficult for antivirus signatures to stop these infestations. Additionally, malware authors create a plethora of dangerous efforts every day, using a variety of techniques such as code insertion and obfuscation code to accomplish their goals.

The creators of these harmful programs use toolkits with mutating engines and polymorphic packers, commonly referred to as polymorphism engines, to change non-obfuscated malware into polymorphic variations. The malware created using this technique usually enters the victim's computer undetected, giving it the opportunity to accomplish its goals before being discovered. There are two widely used methods for analyzing harmful code: the static method and the dynamic method.

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

A. Static Malware Analysis

Static malware analysis is the process by which security experts look at a malicious program without executing its code. The objective is to identify malware families, their modes of operation, and their capabilities. Since there is no code execution, a live environment is not necessary for static malware analysis. However, this could lead scientists to miss crucial information about the infection that is only apparent when they watch it in action. The following are some particular advantages of static analysis: It's incredibly simple and quick. Analysis is predicated on the malicious software's signature.

B. Dynamic Malware Analysis

Dynamic malware analysis involves running the malware's code in a controlled environment and seeing how it interacts with the system. Analysts can identify the true objectives of the virus and how to prevent its detection through this type of research. Although it may take longer, this approach yields a more accurate and comprehensive report. Furthermore, specialized tools are required, and malware may penetrate the analysis environment. Compared to static analysis, dynamic analysis yields more accurate results.

In order to combat sophisticated evasion strategies, the study proposed by Su, L., Cheng, H., Li, L., Zhang, C., Wang, Y., & Zhao, J. (2024), introduces a novel method for ransomware detection called Dynamic Obfuscation Signature Analysis (DOSA), which blends machine learning with static and dynamic analysis. When it comes to polymorphic ransomware, which constantly modifies its code to evade detection, conventional techniques like signature-based detection frequently fall short. In order to overcome this constraint, DOSA updates its detection system in real time and maps obfuscation

patterns dynamically. The findings demonstrate that DOSA maintains a low false positive rate (1.8%), achieves excellent accuracy (92.3% to 97.8%), and swiftly adjusts to new ransomware variations. Its multi-layered architecture guarantees thorough examination of ransomware detection that is file-, network-, and memory-based. Even though DOSA shows notable advancements over traditional methods, problems still exist in identifying highly obfuscated ransomware and maximizing computational effectiveness for real-time applications. The study highlights the importance of adaptive and evolving security solutions in combating modern ransomware threats, making DOSA a valuable advancement in cybersecurity.[4]

In their publication, Arlowe, M., Fitzpatrick, D., & Brannon, C. (2024), the study offers a sophisticated technique for identifying polymorphic ransomware that combines evolutionary algorithms with Generative Adversarial Networks (GANs). The model's capacity to identify novel and highly obfuscated threats is enhanced by the constant generation and improvement of ransomware samples. The study illustrates how adversarial training can improve detection accuracy while highlighting the drawbacks of conventional static and heuristic-based detection techniques. With low false positive rates (less than 5%) and excellent accuracy (96.2%) and recall (94.7%), the results demonstrate that this strategy is more successful than current models. The method's real-time use may be limited by its high computing requirements, which include 12.5 GB of GPU memory and 72 hours of training. There is a need for more advancements because certain extremely complex ransomware variants, like Ragnar Locker, continue to avoid detection. While the study provides valuable insights, it lacks discussion on real-world deployment strategies and comparisons with hybrid detection models.[5]

In their publication, Cuzzocrea, A., Quebrado, M., Hafsaoui, A., & Serra, E. (2023, June), employed a graph-based technique in which they extracted attributes connected to a control flow graph from Android APK binary files. They employed a unique graph representation learning technique called Inferential SIR-GN for Graph representation, which preserves graph structural similarity, in conjunction with a well-known machine learning model called XGBoost to handle the resulting graph. The method is then used to MALNET, an open cybersecurity database with 1,262,024 million Android APK binary files in total, categorized into 47 kinds and 696 families. The experimental findings show that their graph-based approach works better in terms of detection accuracy than the image-based approach. [6]

In order to identify polymorphic malware, Deng, X., & Mirkovic, J. (2022, January) recommend using clustering algorithms on malware network traffic patterns. The authors demonstrate how a sizable number of clusters are produced by grouping data units and byte frequency inside the malware's network traffic. The samples in these clusters have the same local characteristics and very similar flows. Their method, which is based on embedding patterns, produces more than 90% of potentially polymorphic clusters and up to 80% of genuinely polymorphic clusters, whereas the use of local DLL (Dynamic Link Library) access patterns for identifying malware samples that are polymorphic in nature has limited effectiveness due to the widespread reuse of common DLL files. Furthermore, this approach provides human-interpretable network patterns that facilitate understanding the malware's fundamental characteristics. [7]

The study "Deceiving AI-based Malware Detection through Polymorphic Attacks" examines how Convolutional Neural Networks (CNNs) are vulnerable to malware detection, specifically when it

comes to polymorphic malware that changes to avoid detection. The paper demonstrates how adversarial machine learning approaches can be used to trick CNN-based malware classifiers, which rely on image-based analysis of binary executables. The study presents a polymorphic attack technique in which malicious software is altered to look like harmless software while maintaining its harmful capabilities. The findings show that CNNs have 100% evasion rates against some polymorphic malware samples, even though they have a high accuracy rate in classical malware detection. The study investigates adversarial training, which is retraining CNNs using adversarial cases to increase robustness, as a means of combating this. But even after using this method, evasion rates can still be high in specific situations and detection rates can still fluctuate. The study comes to the conclusion that although CNNs have potential malware detection capabilities, hybrid detection techniques should be used in conjunction with them for enhanced cybersecurity because they are extremely vulnerable to adversarial attacks.[8]

Akhtar et al. (2022) used machine learning techniques to address the malware identification challenge. The authors of the study suggested a protection strategy that involved evaluating the effectiveness of three distinct machine learning algorithms for detecting malicious software in order to select the most effective one. Notably, all three of these approaches performed well in terms of detection accuracy. According to the study's findings, the machine learning algorithms taken into consideration in this inquiry were able to differentiate between data that was hazardous and data that was not. The Decision Tree (DT) machine learning approach stood out among the assessed classifiers with the highest correctness rate of 99%. According to the experiment's findings, static analysis—which is focused on data from Portable Executable (PE) files and carefully chosen data—showed a great deal of promise for offering precise malware characterizations and perhaps the highest level of detection accuracy. [6]

By employing feature selection techniques, Selamat and Ali (2022) came to a noteworthy result and showed that they were able to identify traits specific to polymorphic malware. The study illustrated how crucial it is to select the most pertinent attributes, since not all of the gathered features were shown to be advantageous for ML classifiers. This study's feature selection method, which was based on a deep comprehension of the PE32 header structure, demonstrated an impressive level of detection accuracy, ranging from 90% to 100%. This demonstrates how these techniques may be used for quick and efficient detection. [7]

A brand-new machine learning-based classification method known as New Feature Manufacturing i.e. Engineering (NFM i.e. NEF) was presented by Masabo and associates in 2019. NFE uses the traits and actions of malware to categorize and detect malware that is polymorphic. This method achieves an impressive 94% accuracy in malware cataloguing through a three-step process that includes K-nearest neighbors (KNN), linear analysis, and the Gradient Boosting machine. [8]

In a different investigation, Ojugo Team (2019) used a string matching algorithm called the Boyer-Moore approach to find the dangerous program. This method prioritizes non-matching characters while performing a linear analysis of the string to find matching characters. The length of the string and the character's location must be known in order to use this approach. [9]

In a recent study, Vinayakumar and colleagues (2019) examined the effectiveness of machine learning algorithms (MLAs) in malware detection. The study's conclusions show that, especially when dealing

with malware variants, the traditional approach of evaluating malware behavior by combining static and dynamic analysis can be time-consuming and ultimately inadequate. Consequently, the study proposed deep learning, a more advanced machine learning method. This novel strategy used two different image processing techniques. Using a new framework called ScaleMalNet, which uses deep learning, the researchers collected malware samples from end users and then processed them in two steps. Malware is first grouped into matching malware categories using a combination of dynamic and static analysis approaches, followed by image processing. [10]

The paper "Malware Classification Using API System Calls" explores the effectiveness of data mining techniques in classifying malware based on API system calls. Traditional malware detection methods, such as signature-based and heuristic approaches, struggle with polymorphic and metamorphic malware due to their ability to change structure while maintaining functionality. To address this, the study analyzes 4-gram API call sequences extracted from 552 Windows PE files using the Cuckoo sandbox. The extracted features were processed with TF-IDF and classified using Support Vector Machines (SVM), Decision Trees, Random Forest, and Gaussian Naive Bayes. The results show that SVM and Decision Trees performed best, achieving accuracy between 92% and 96.4%, while Gaussian Naive Bayes had the lowest performance. The study highlights that classifying malware based on behavioral API calls offers a more robust detection method, as malware execution remains consistent despite obfuscation techniques. However, the paper notes challenges in detecting backdoors and Trojans, which are sometimes misclassified due to their resemblance to benign software. Overall, this research demonstrates that API-based classification is an effective approach for malware detection and classification, with potential for further improvements through larger datasets and cross-platform evaluations.[14]

In an improved version of earlier work, Drew and colleagues (2017) created a novel approach to detect malicious software that is based on polymorphism by employing ensemble approaches and sequence classification algorithms [11][15]. The 500 GB Kaggle dataset from the 2015 Microsoft challenge was used in their study, and the STRAND algorithm helped with the cataloguing work. They combined the models generated from the asm and bytes data using the STRAND algorithm to produce ensembles, utilizing Microsoft's bytes (.bytes) and assembly (.asm) data properties. They used the minihash approach to determine commonalities. The overall detection accuracy reached a remarkable 98% when the minihash scores of the produced models were calculated. [12]

In their study, Tong and Yan (2017) introduced a unique hybrid approach for mobile malware detection that outperformed current techniques in detecting various mobile app threats with higher accuracy rates. When compared to other currently used technologies, our method is more accurate, efficient, and precise in identifying malware. The approach is straightforward and relies on cost-effective algorithms to handle the data. It does have a disadvantage, though, in that in order to maintain detection accuracy, fresh malware variants must be continuously acquired along with safe application samples. Furthermore, mobile devices may not be able to manage massive volumes of data due to their limited CPU and storage capabilities. [13]

In order to detect polymorphic and metamorphic malware, the study "An Advanced Approach to Polymorphic/Metamorphic Malware Detection using Hybrid Clustering Approach" suggests a hybrid approach that combines pattern matching and signature-based techniques. These malware varieties are

difficult for traditional signature-based techniques to combat since they can change the code structure without affecting functionality. In order to categorize malware variants, the study presents a clustering-based detection model that examines system calls and executed instructions. To increase classification accuracy, the K-Nearest Neighbors (KNN) algorithm is used to differentiate between malware that is polymorphic and malware that is metamorphic. The sliding window protocol used in the experimental setup improves detection performance and malware classification accuracy. [18]

The study showed that their proposed detection method was effective in detecting network intrusions. In this context, Ambusaidi and colleagues (2016) introduced a novel filter-based feature selection approach along with a theoretical examination of mutual information to assess the interdependence between features and output classes. FMIFS, a versatile approach developed to tackle feature selection challenges, was introduced. The feature selection methods presented in this learning are good at ranking features based on their significance, but they don't figure out the ideal number of features needed for training a classifier. The proposed algorithm for feature selection could be further improved by improving the search method [14].

Atici and their group (2016) shed light on the methods used to find software that is polymorphic and its different variations. By applying a set of filters, correcting any unintentional exclusions, and modifying the process to include these samples, the researchers described a method for finding samples. The system's filters are updated automatically to guarantee quick handling of new polymorphic malware iterations and to avoid their misclassification. A number of variables influence the choice of filters for evaluating polymorphic samples, including dispositive scans and the characteristics of the exe (i.e., executable) file. Before attempting to match the signature of the malicious polymorphic variant, a section of the exe file is decrypted as part of the process during the encryption scan. [15]

Fraley et al. (2016) presented a detection strategy that uses both static and dynamic analysis techniques to extract topological features. The suggested approach uses IDA Pro and Cuckoo Sandbox for feature extraction, leveraging data mining techniques to expedite the classification process. A total of 3637 samples—2400 clean samples, 800 malicious samples, and 437 unlabeled samples—were used in their research. By examining function call graphs, the researchers tracked usage patterns of instructions and used belief propagation to identify traits of compromised files. A thorough feature set was produced by carefully selecting and utilizing twenty features, which included eight behavioral and twelve structural properties. File size, compiler type, and MOV, ADD, and LEA instructions were among the features that were extracted. The final feature set was transformed into an ARFF data format that the Weka data mining application could use. A 10-fold cross-validation technique was used for validation, and Weka was used for classification. The use of ensemble bagging techniques produced an outstanding overall accuracy of 99.9% and a low false negative rate of 0.001[16].

3. Discussion

- Researchers have used a variety of approaches to address the problem of polymorphic malware, which has been extensively studied in scholarly literature. Bioinformatics DNA approaches including Needleman-Wunsch, Smith-Waterman, Multiple Sequence Alignment, and the Strand gene sequence classifier have been used in several investigations. At the same time, other studies

have dabbled in using machine learning techniques. Various algorithms have been used to create models in some research projects, and the most dependable and efficient method has been chosen. Table 1 provides a comparison of different analysis and detection techniques for polymorphic malware, focusing on methodology, dataset, performance, findings, and challenges.

Highlights from the literature review include:

1. Trojans, worms, and viruses are examples of polymorphic malware, which may encrypt, decode, and change the appearance of data through code propagation. Therefore, merely using a signature-based method does not ensure their identification.

2. Using a variety of machine learning classifiers and mining algorithms results in remarkably high accuracy rates. This method, which focuses on Windows API calls, creates opportunities for further research on a variety of obfuscation-related topics.

3. Although the sandbox approach to analysis provides dynamic tools, its effectiveness is restricted since malicious programs can detect virtualization frameworks by examining Internet access. However, it turns out to be a dependable technique that guarantees an accurate and good detection rate when paired with analysis based on static nature.

4. The vast number of new malicious software types that are always appearing on a daily basis outstrips the capability of the detection techniques that are now in use.

Table 1. Comparative Analysis of Polymorphic Malware Detection Techniques

Author(s) & Year	**Methodology**	**Dataset Used**	**Performance Metrics**	**Key Findings**	**Limitations & Challenges**
Su et al. (2024)	Dynamic Obfuscation Signature Analysis (DOSA) combining static, dynamic analysis & machine learning	Multiple ransomware samples from various sources	Accuracy: 92.3%-97.8% False Positive Rate: 1.8%	Real-time adaptation to new ransomware, low false positive rate, multi-layered detection approach (file-based, network-based, memory-based).	Struggles with highly obfuscated ransomware; requires optimization for real-time detection and computational efficiency.
Arlowe et al. (2024)	Generative Adversarial Networks (GANs) with evolutionary algorithms for detecting polymorphic ransomware	Large dataset of polymorphic ransomware samples	Accuracy: 96.2% Recall: 94.7% False Positive Rate: <5%	Outperforms traditional models in detecting obfuscated ransomware; improves recall and detection of novel threats.	High computational cost (72 hours training, 12.5 GB GPU usage); lacks discussion on real-world deployment strategies.

Cuzzocrea et al. (2023)	Graph-based detection using control flow graphs and XGBoost	MALNET Dataset: 1,262,024 Android APK files	Higher accuracy than image-based malware detection	Graph-based learning significantly improves malware classification over image-based methods.	Limited to Android malware detection; may not generalize to other platforms.
Deng & Mirkovic (2022)	Clustering techniques on malware network traffic patterns	Large dataset of malware network traffic flows	90% potentially polymorphic clusters 80% genuinely polymorphic clusters	Clustering helps detect polymorphic malware based on network behavior patterns.	Common DLL usage limits detection effectiveness; requires more robust pattern extraction.
Akhtar et al. (2022)	Comparative evaluation of machine learning algorithms for malware detection (Decision Tree, SVM, Random Forest)	Dataset of malware samples	Accuracy: 99% (Decision Tree)	Decision Tree (DT) outperforms other ML models for static malware detection.	Static analysis struggles against obfuscated and metamorphic malware.
Selamet & Ali (2022)	Feature selection for polymorphic malware detection based on PE32 header format	Malware dataset with PE32 headers	Accuracy: 90%-100%	Selecting relevant features improves ML classifier performance in detecting polymorphic malware.	Feature selection requires refinement for handling large datasets.
Masabo et al. (2019)	New Feature Engineering (NFE) using KNN & Gradient Boosting	Malware classification dataset	Accuracy: 94%	ML-driven feature engineering improves polymorphic malware classification.	Real-time applicability not tested; needs evaluation in dynamic malware environments.
Ojugo & Eboka (2019)	Boyer-Moore string matching algorithm for	Various malware string datasets	String matching-based malware	Efficient string-based malware detection	Requires predefined string length, limiting

	malware detection		identification		flexibility against unknown threats.
Vinayakumar et al. (2019)	Deep learning with image processing (ScaleMalNet)	Large dataset of malware images	Improved classification accuracy over traditional ML	ScaleMalNet improves malware detection using deep learning-based image processing.	Computationally expensive; not feasible for resource-limited environments.
Drew et al. (2017)	Ensemble learning with STRAND algorithm	Microsoft Kaggle Dataset (500GB)	Accuracy: 98%	Combines assembly (.asm) and byte (.bytes) data for improved classification.	Focused only on Windows malware; lacks applicability to other platforms.
Tong & Yan (2017)	Hybrid approach for mobile malware detection	Mobile malware dataset	High accuracy for mobile threats	More precise than traditional mobile malware detection techniques.	Requires continuous updating of malware samples; computational cost limits mobile deployment.
Ambusaidi et al. (2016)	Feature selection using FMIFS for network intrusion detection	Intrusion detection system dataset	Improved feature ranking for classification	FMIFS enhances feature selection for better intrusion detection.	Lacks an optimal method for determining the right number of selected features.

4. Conclusion

The most recent methods used to identify dangerous polymorphic software are highlighted in this study. Internet-borne malware is the main security concern that the field of digital security faces on a daily basis. People's physical security as well as their priceless possessions, such as data and technology, are at risk from these attacks. The complexity of contemporary malware, which uses polymorphic tactics, makes it difficult for current technology to effectively handle this problem. Many solutions rely on signatures, but as malware is always evolving, signatures must also be updated. The detection rate should ideally be 100%, yet there are challenges due to high False Positive and False Negative Rates.

Regularly updating signature-based systems requires a lot more resources. According to earlier research, between 25% and 50% of malware is currently detected by anti-malware software. Eliminating these hazards is difficult due to their secretive character and the difficulty of classifying them within their own families. Because defenses rely on signatures, polymorphic malware is

essentially invisible. Currently, polymorphic malware has an effective correction rate between 68.75% and 81.25%.

Currently, polymorphic malware cannot be found or identified at an endpoint using a single technique. Advanced malware affects almost every business, government agency, and institution with a network and internet connection. The dynamic nature of polymorphic malware causes traditional signature-based detection methods to fail. Multiple distribution methods, including email, embedded files, software updates, and websites, are used by polymorphic malware to propagate. Large security companies spend billions of dollars fighting malware in general, but identifying malicious-polymorphic software is made more difficult by the expertise of malware writers. Malicious software developers are constantly improving their programs to avoid being discovered by security monitoring systems. There is no one-size-fits-all method for identifying dangerous polymorphic and malignant metamorphic software; instead, continuous malware analysis is required to comprehend how these threats are changing.

References

[1] Jovanovic, Bojan. "A Not-So-Common Cold: Malware Statistics in 2024." A Not-So-Common Cold: Malware Statistics in 2024, DataProt, February 06,2024, https://dataprot.net/statistics/malware-statistics/. Accessed February 06,2024

[2] AV-TEST. "Total amount of Malware and PUA." Malware, 2024, https://www.av-test.org/en/statistics/malware/.

[3] Chaikovskyi, M., Chaikovska, I., Sochor, T., Martyniuk, I., & Lyhun, O. (2024). Comprehensive approach to the detection and analysis of polymorphic malware. In CEUR Workshop Proceedings (Vol. 3736, pp. 312-323).

[4] Su, L., Cheng, H., Li, L., Zhang, C., Wang, Y., & Zhao, J. (2024). A novel approach of ransomware detection with dynamic obfuscation signature analysis.

[5] Arlowe, M., Fitzpatrick, D., & Brannon, C. (2024). Evolutionary Generative Adversarial Networks for Detecting Novel Polymorphic Ransomware

[6] Cuzzocrea, A., Quebrado, M., Hafsaoui, A., & Serra, E. (2023, June). A Graph-Representation-Learning Framework for Supporting Android Malware Identification and Polymorphic Evolution. In 2023 10th IEEE Swiss Conference on Data Science (SDS) (pp. 34-41). IEEE.

[7] Deng, X., & Mirkovic, J. (2022, January). Polymorphic malware behavior through network trace analysis. In 2022 14th International Conference on Communication Systems & Networks (COMSNETS) (pp. 138-146). IEEE.

[8] Catalano, C., Chezzi, A., Angelelli, M., & Tommasi, F. (2022). Deceiving AI-based malware detection through polymorphic attacks. Computers in Industry, 143, 103751.

[9] Akhtar, M. S., & Feng, T. (2022). Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry, 14(11), 2304

[10] Selamat, N. S., & Ali, F. H. M. (2022). Polymorphic Malware Detection based on upervised Machine Learning. Journal of Positive School Psychology, 6(3), 8538-8547.

[11] Masabo, E. (2019). A Feature Engineering Approach for Classification and Detection of Polymorphic Malware using Machine Learning (Doctoral dissertation, Makerere University).

[12] Ojugo, A., & Eboka, A. O. (2019). Signature-based malware detection using approximate Boyer Moore string matching algorithm. International Journal of Mathematical Sciences and Computing, 5(3), 49-62
[13] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Robust intelligent malware detection using deep learning. IEEE Access, 7, 46717-46738.
[14] Ninyesiga, A., & Ngubiri, J. (2018). Malware classification using API system calls. International Journal of Technology and Management, 3(2), 9-9
[15] Drew, J., Moore, T., & Hahsler, M. (2016, May). Polymorphic malware detection using sequence classification methods. In 2016 IEEE Security and Privacy Workshops (SPW) (pp. 81-87). IEEE.
[16] Drew, J., Hahsler, M., & Moore, T. (2017). Polymorphic malware detection using sequence classification methods and ensembles. EURASIP Journal on Information Security, 2017(1), 1-12.
[17] Tong, F., & Yan, Z. (2017). A hybrid approach of mobile malware detection in Android. Journal of Parallel and Distributed computing, 103, 22-31.
[18] Sharma, P., Kaur, S., & Arora, J. (2016). An advanced approach to polymorphic/ metamorphic malware detection using hybrid clustering approach. International Research Journal of Engineering and Technology, 3(6), 2229-2232.
[19] Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. IEEE transactions on computers, 65(10), 2986-2998.
[20] Atici, M. A., Sagiroglu, S., & Dogru, I. A. (2016, April). Android malware analysis approach based on control flow graphs and machine learning algorithms. In 2016 4th International Symposium on Digital Forensic and Security (ISDFS) (pp. 26-31). IEEE.
[21] Fraley, J. B., & Figueroa, M. (2016, March). Polymorphic malware detection using topological feature extraction with data mining. In SoutheastCon 2016 (pp. 1-7). IEEE.
[22] Pawar, J. A., Avhankar, M. S., Gupta, A., Barve, A., Patil, H., & Maranan, R. (2024, May). Enhancing network security: Leveraging isolation forest for malware detection. In 2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT) (pp. 230-234). IEEE.
[23] Avhankar, M. S., Pawar, J., Singh, G., Asokan, A., Kaliappan, S., & Purohit, K. C. (2023, May). Simulation Environment for the I9 Vanet Platform. In 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) (pp. 1-8). IEEE.
[24] Avhankar, M., & Pawar, D. J. (2023). An Experimental Comparison of AODV and DSDV Routing Protocols in MANET. *International Journal of Scientific and Research Publications*, *13*(3).