Journal of Informatics Education and Research
ISSN: 1526-4726
Vol 5 Issue 3 (2025)

Comprehensive Evaluation on Computational Intelligence Techniques for Malware Analysis

Deepak Prashar [1*], Bhuvan Unhelkar[2], Sandeep Ranjan[3]

[1]Lovely Professional University, Phagwara, India deepak.prashar@lpu.co.in
[2]University of South Florida, Tampa, USA bhuvan.unhelkar@gmail.com
[3]CTIEMT, Jalandhar, India ersandeepranjan@yahoo.com

Abstract: -
Malicious software has constituted threats to system security and operations carried out online due to the Internet and other technologies and associated breaches. In the past, scholarly research focused on the use of classical methods of malware detection and associated attributes like signature and heuristics, etc. Conventional malware detection methods become ineffective against emerging malware versions and their advanced evasion strategies. Lately, there have been attempts to put to use sophisticated Machine Learning (ML) for malware detection due to traditional techniques being highly ineffective on emerging variants of malware. This paper focuses on and implements methods for detecting and identifying the malware's features using hash values associated with the Virus Total and the Quick Heal dataset. Also, this research surveys the recent ML techniques in malware detection and focuses on ML methods' sophistication and parameters like precision, recall, and the used datasets and methods. In addition, this research identifies in depth the highly understudied area of malware analysis.

Keywords: Malware, Machine learning, Deep learning, Random Forest, Logistic regression, Convolution neural network

1. Introduction

The increasing demand for personal computers and internet connectivity is one of the prime reasons for the continuous increase in malware.[1] On account of business and normal tasks in the organizations are more centric towards the usage of personal computers and internet, malicious software writers have a wider scope to exploit weaknesses and expose system vulnerabilities. The extensive use of the internet leads to the interlinking of millions of devices, making the rapid dissemination of malware. Cybercriminals may disseminate malware via vulnerable websites, email attachments, and social media, among other channels. The rising incidence of malware is a significant security concern, requiring ongoing research into effective detection methods. It essentially includes two approaches for malware detection: signature-oriented identification and behavior-oriented identification. The signature dependent is rapid and effective just for detecting recognized malware; however, the behavior-based method may identify unknown and intricate malware to a certain degree using machine intelligence and other methodologies; nevertheless, the behavior-based method is inherently difficult. No approach can identify all types of malwares, particularly when the prevalence of malware increases daily. The signature-based methodology generates a distinctive signature based on the fundamental object's characteristics. The method effectively detects the existence of a digital signature by scanning the item. The behavior-based approach evaluates the planned activities of objects before their execution. Previous malware was easily detectable due to its ability to conceal its characteristics; however, the current malware employs many tactics, such as obfuscation [2-5], to obscure its identity for extended periods and can even circumvent firewalls and other security measures inside the network or system. Moreover, the attack employs multiple malware types, leading to even more devastating outcomes.

The internet has become a hotspot for all kinds of security threats with no signs of stopping. This has included cybercrimes such as malware which has become a huge threat as it has the ability to disable the CIA of the compromised system as well as steal sensitive data such as passwords, contacts, and bank information. Difficulty in the detection and the examination of malware arises from the multiple and overlapping layers of complex deflective methods, the inaccuracy of algorithms, and the absence of the traditional methods dealing with novel malware threats.[6] The objective is to analyze harmful samples to get a thorough knowledge of their behavior and the changes they experience over time.[7] Multiple domains such as program and network evaluation are integrated to determine harmful samples to understand the malware behaviors and transformations over time. Malware poses a major threat to security, and research to improve detection is essential. There are two fundamental methods of malware detection. The first is signature-based recognition, and the second is behavior-based recognition.[8]

The substantial rise in malware attacks has prompted the creation of advanced malware identification techniques to safeguard the present infrastructure and alleviate the financial loss on account of these attacks. Cyber infrastructures have widely relied on Machine Learning (ML) techniques to protect against malware attacks.[9-10] Consequently, machine learning models have shown considerable effectiveness in tackling malware detection issues, and several big corporations worldwide have used these models to identify malware assaults on their systems.[11] Despite the many benefits offered by contemporary classical ML models, their effectiveness and success mostly depend on human selection of suitable variables for training and evaluation. This technique is laborious and requires significant time commitment.[12] Furthermore, ML methodologies are inadequate for handling large datasets.[13] But, the creation of Deep Learning (DL) methods addressed the bottlenecks observed among ML models. DL algorithms excel at handling extensive datasets and executing autonomous feature fetching and finalization.[14] Consequently, building a malware classification system with traditional ML approaches requires knowledge of malware feature collection methods and model feature architecture overview needed to classify or detect novel input. The antivirus software remains vital to the system, securing system, resource, and information access from malware. As of now, malware creators use advanced methods to hide malware, including obfuscation, packing, encoding, and even cryptographic techniques. Because of these advanced methods, most traditional malware detection systems struggle to identify the new malware variants.

This study seeks to provide a thorough and current review of prevailing developments concerning ML and other related methodologies used for the analysis, detection, and classification of malware. Furthermore, we provide some insights and recommendations for future directions.

The following sections, each providing detailed information, comprise the paper: The 2nd section discusses the necessity of malware analysis, while the 3rd section details the process of utilizing the Virus Total application for analysis. The 4th section defines the adopted methodology, while Section 5th illustrates the related work. The 6th section conducted a comparative evaluation of the studied work using the parameters, while the 7th section covers the open research issues. The final section concludes with a focus on future work.

2. Malware Analysis and Its Need

Malware analysis is the study of malicious software to understand how it operates, how it behaves, and the likely impacts of the malicious software. In part, malware analysis encapsulates the capture and documentation of malware's deeply intricate workings, detecting the minutiae of every encoded byte that is passing information or manipulating the original source code that an intruder has forged. The analysis also highlights the capture of malware's critical attributes which informs the study of malware's functional behaviors and how it operates within a given system or network. There are three fundamental methods one can use to assess and identify malware, which are, static, dynamic, and hybrid. Each serves an important purpose and assists in different ways in the overall malware assessment process.

Static evaluation works by examining an executable document's architecture without processing them. The executed file contains several static features, like unique components and memory efficiency. It comprises two components: fundamental and advanced. Fundamental static analysis emphasizes the essential qualities and attributes of malware in order to gain a first understanding of its features. During fundamental static analysis, various techniques enable the acquisition and analysis of file dimensions, file category, and header details. After conducting a fundamental static evaluation, one may employ sophisticated static evaluation approaches to have a detailed understanding of the malware's actions and attributes. Static assessment entails a thorough review of the software instructions.[15] However, it encounters the issue of not recognizing fresh polymorphic malware which have the tendency to overcome static evaluation. Static evaluation has challenges in detecting concealed malware due to its inefficiency in examining packed samples.

Dynamic evaluation entails executing program instructions and scrutinizing malware behavior. To protect the system against malware, dynamic evaluation is performed in a monitored scenario such like sandbox. It is classified into two distinct groups: basic and advanced. In basic one, tools are employed to understand the working of malware. In contrast, advanced dynamic evaluation utilizes tools. Such tools enable users to perform particular commands while altering arguments and variables.[15] In dynamic evaluation, the program operates within an environment that provides it unrestricted access to every single resources. [16-17] It can tackle hiding of malwares and identify latest types.

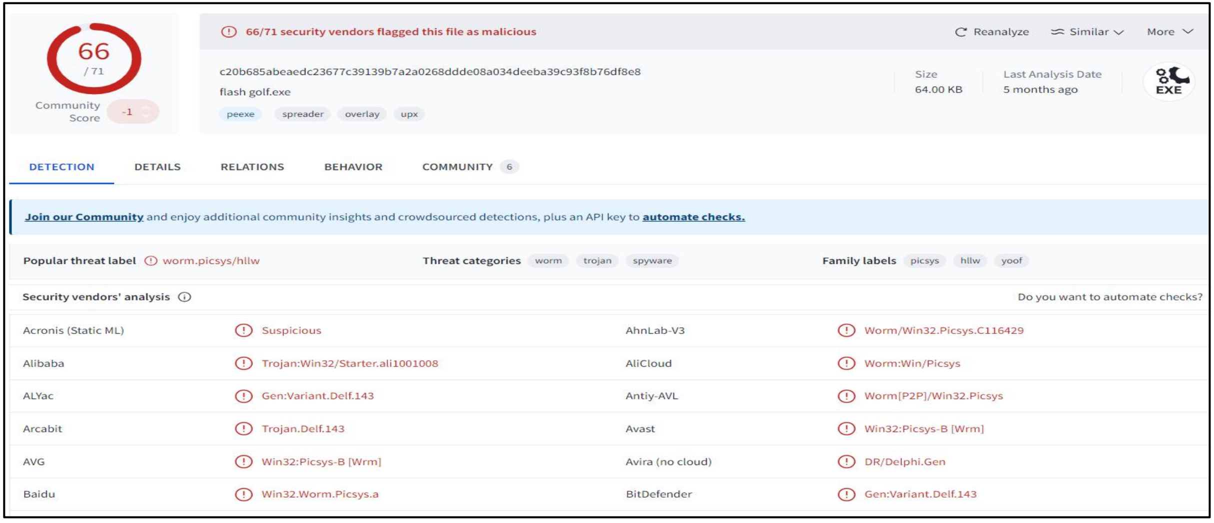
Hybrid evaluation is a methodology that uses combination of dynamic and static techniques to detect and analyze malware. This approach begins with static analysis of the malware where the source code and structure are analyzed without executing the code. It then incorporates a dynamic approach for a more complete analysis. The use of dynamic analysis within a hybrid approach alleviates the drawbacks associated with only using the other two forms of analysis. It enables a more complete understanding of the malware's intricacies. The combinational use of the two forms of analysis streamlines the analysis, improving the effectiveness of the identification and assessment of the malware.[18-19]

The importance of classifying malware accurately is equivalent to the importance of detecting it. Malware comes in several types: viruses, Trojan horses, worms, rootkits, ransomware, and keyloggers. There are different approaches to malware classification, including the use of feature-based algorithms and image processing.[20] As with most things, having more information would be beneficial to improve classification. Constructing powerful classifiers using advanced machine intelligence will aid in more accurate malware identification to improve classification. [21-22] Identification and removal of malware is critical to prevent unwanted use and information loss, and to protect software and hardware against potential loss and damage. [23] Malware usually aims to capture sensitive information, such as banking credentials, which can lead to unauthorized purchases and identification theft which can be very financially damaging. By detecting and removing malware, the user is protected against potential financial loss, which is very important. Malware can also take sensitive information, thus compromising user confidentiality.

3. Malware Analysis Using Virus Total

The analysis of malware has many valid reasons. Helping users figure out if a file in question is a threat is one of them. In the case of a positive threat assessment, users can learn ways of threat mitigation for their networks and IT systems. In the case of a security breach, malware analysis can help ascertain the damage in totality. In some instances, malware authors hide clues about their identity in the malware itself. Consider a situation in which there is a suspicious file in the user environment. It might be contained in a security appliance, like a web proxy or an email gateway. Even though the file is ostensibly harmless because no antivirus software flagged it, it might still be suspicious. In that case, a specific procedure should be followed that includes the generation of the file hash followed by a verification of the file on Virus Total. [24] A file hash is a unique sequence of characters that corresponds to a specific file. This is that file's fingerprint which is of considerable importance because the file name changes during every infection. Cryptographic hashing methods like MD5 (Message Digest 5) or SHA (Secure Hash Algorithm) ensures that a hash value will be created which gives a file an identification string of a specific length. Virus Total is a great resource for analyzing a possibly harmful file. It lets a user submit file for evaluation against multiple antivirus scanners. It stores the result of to the previous queries. This allows user perform a search against a large dataset. We used the SHA1 hash to perform a search for the file in question.

In this instance, we gathered a dataset using Quick Heal technologies, containing the hash values of approximately 108 malicious types. The dataset is used to analyze the detailed information about the particular malware using the Virus Total. [25]



66/71 security vendors flagged this file as malicious

c20b685abeaedc23677c39139b7a2a0268ddde08a034deeba39c93f8b76df8e8

flash golf.exe

peexe spreader overlay upx

Size 64.00 KB

Last Analysis Date 5 months ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label worm.picsys/hllw

Threat categories worm trojan spyware

Family labels picsys hllw yoof

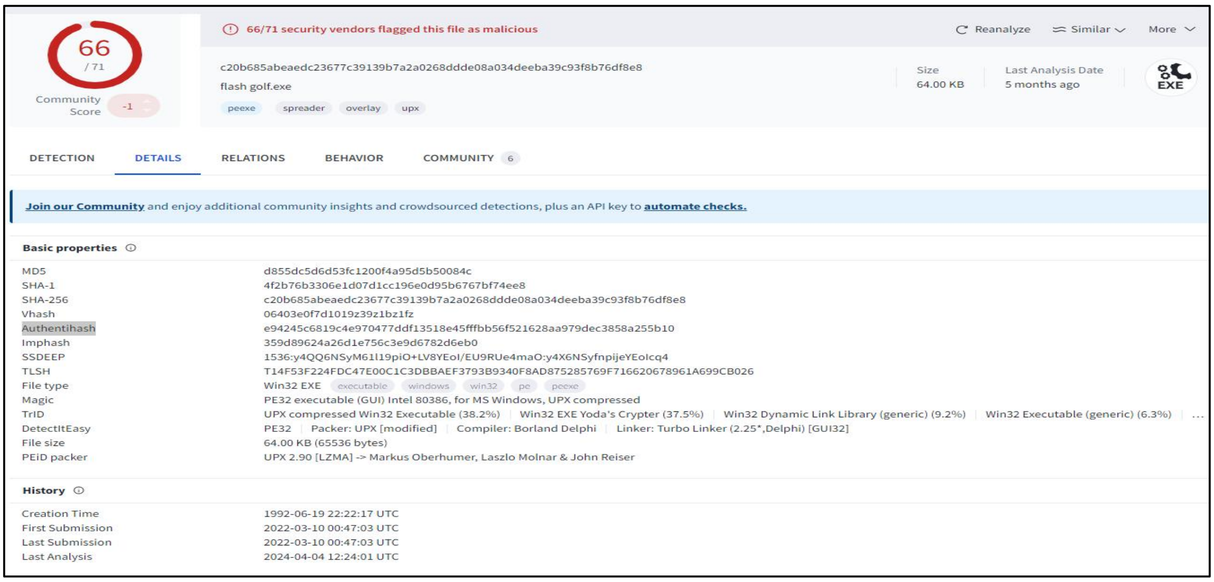
Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Suspicious	AhnLab-V3	Worm/Win32.Picsys.C116429
Alibaba	Trojan:Win32/Starter.ali1001008	AliCloud	Worm:Win/Picsys
ALYac	Gen:Variant.Delf.143	Antiy-AVL	Worm[P2P]/Win32.Picsys
Arcabit	Trojan.Delf.143	Avast	Win32:Picsys-B [Wrm]
AVG	Win32:Picsys-B [Wrm]	Avira (no cloud)	DR/Delphi.Gen
Baidu	Win32.Worm.Picsys.a	BitDefender	Gen:Variant.Delf.143

Figure 1: Analysis of suspicious file by Virus Total

Figure 1 suggests that 66 out of 71 antivirus engines classified the file as malware. This suggests that the file is classified as malware, as the majority of antivirus providers share this tendency. Virus Total presents the malware classification labels from many antivirus providers. The findings indicated that the file was not only harmful, but also appeared to be some type of worm.



66 / 71

Community Score -1

66/71 security vendors flagged this file as malicious

Reanalyze Similar More

c20b685abeaedc23677c39139b7a2a0268ddde08a034deeba39c93f8b76df8e8

flash golf.exe

peexe spreader overlay upx

Size 64.00 KB

Last Analysis Date 5 months ago

EXE

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 6

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Basic properties

MD5	d855dc5d6d53fc1200f4a95d5b50084c
SHA-1	4f2b76b3306e1d07d1cc196e0d95b6767bf74ee8
SHA-256	c20b685abeaedc23677c39139b7a2a0268ddde08a034deeba39c93f8b76df8e8
Vhash	06403e0f7d1019z39z1bz1fz
Authentihash	e94245c6819c4e970477ddf13518e45fffbb56f521628aa979dec3858a255b10
Imphash	359d89624a26d1e756c3e9d6782d6eb0
SSDEEP	1536:y4QQ6NSyM61l19piO+LV8YEoI/EU9RUe4maO:y4X6NSyfnpijeYEoIcq4
TLSH	T14F53F224FDC47E00C1C3DBBAEF3793B9340F8AD875285769F716620678961A699CB026
File type	Win32 EXE executable windows win32 pe peexe
Magic	PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
TrID	UPX compressed Win32 Executable (38.2%) | Win32 EXE Yoda's Crypter (37.5%) | Win32 Dynamic Link Library (generic) (9.2%) | Win32 Executable (generic) (6.3%) | ...
DetectItEasy	PE32 | Packer: UPX [modified] | Compiler: Borland Delphi | Linker: Turbo Linker (2.25*,Delphi) [GUI32]
File size	64.00 KB (65536 bytes)
PEiD packer	UPX 2.90 [LZMA] -> Markus Oberhumer, Laszlo Molnar & John Reiser

History

Creation Time	1992-06-19 22:22:17 UTC
First Submission	2022-03-10 00:47:03 UTC
Last Submission	2022-03-10 00:47:03 UTC
Last Analysis	2024-04-04 12:24:01 UTC

Figure 2: Analysis of suspicious file properties and history by Virus Total

Virus Total provides additional information that helps with the evaluation. The Details and Basic Properties tabs offer further information, including the file type as shown in figure 2. Previously, we determined the SHA1 hash of the file; there are other MD5 and SHA256 hash types available. The File Type field also verifies that it is a Windows 32-bit portable executable. On this web page, history is another aspect to examine. This shows the file's previous upload date to Virus Total. This confirms the repeated posting of the suspicious file to Virus Total.

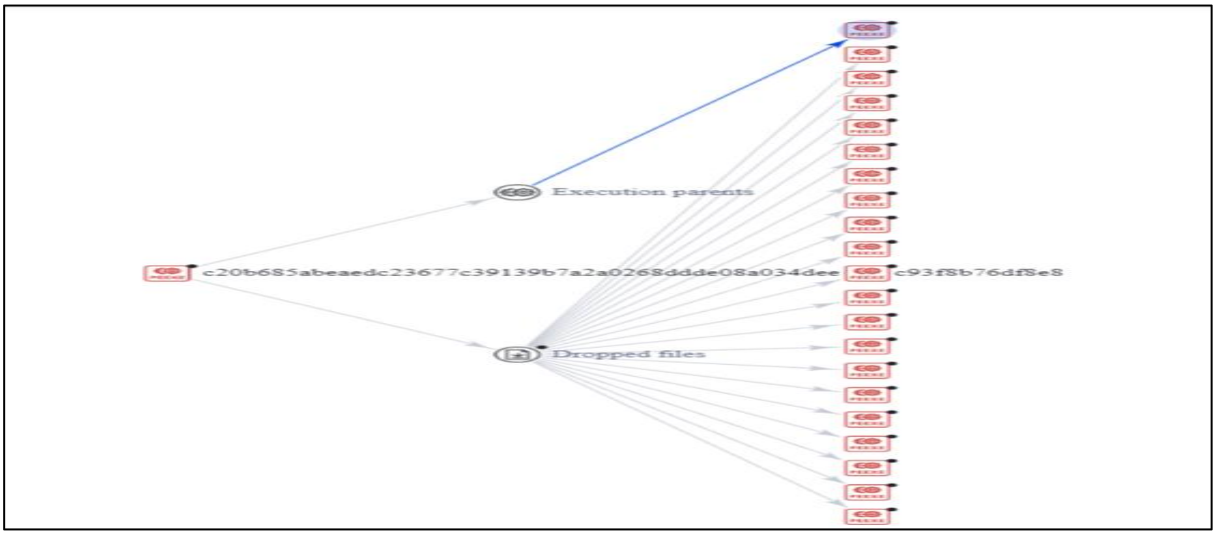


Figure 3: Analysis of threat model for the malware using Virus Total

When examining a malware hash on Virus Total, "execution parents" and "drop files" are essential elements that clarify the order of events culminating in the malware's execution, as depicted in figure 3. This information assists analysts in comprehending the malware's delivery method and identifying any additional malicious files it may have deposited on a system, providing a comprehensive overview beyond the single file under scrutiny. Essentially, they facilitate the identification of a malicious file's "source" and "consequences" by indicating which files instigated its execution, as well as any other files it may have generated during its execution. Similarly, we can examine the other malicious file to extract additional details using Virus Total. In the current scenario, we can use ML and other associated methods to enhance the precision of malware identification and categorization. Therefore, the upcoming section will discuss the previous work conducted in this area by various researchers.

4. Related Methodology

Here, we have elaborated the process that is being followed for executing the comprehensive review pertaining to malware evaluation based on the selection of research articles. We utilize databases like IEEE, Springer, Elsevier, Wiley and Publons to examine important existing content. More focus has been put on the research relevant to malware evaluation using ML. The redundant documents were subsequently eliminated. Furthermore, several manuscripts were rejected for various reasons, including their unavailability as complete works, with just the abstract accessible, thus making them useless. Some documents consist of multiple languages. To ensure the incorporation of contemporary and pertinent research, we have taken the articles available from 2021 till present. Furthermore, precedence has been assigned to publications that concentrated on concepts or aims significant in previous years and hadn't previously been investigated by scholars in earlier periods. The chosen articles mainly highlighted significances pertaining to malware assessment or identification, concentrating on ML-based methodologies for malware evaluation. The whole categorization of the literature work has been presented in the figure 4 below which signifies the prima representation. This signifies that how many papers that are initially identified and then the ones that are finally selected for the review of this work.

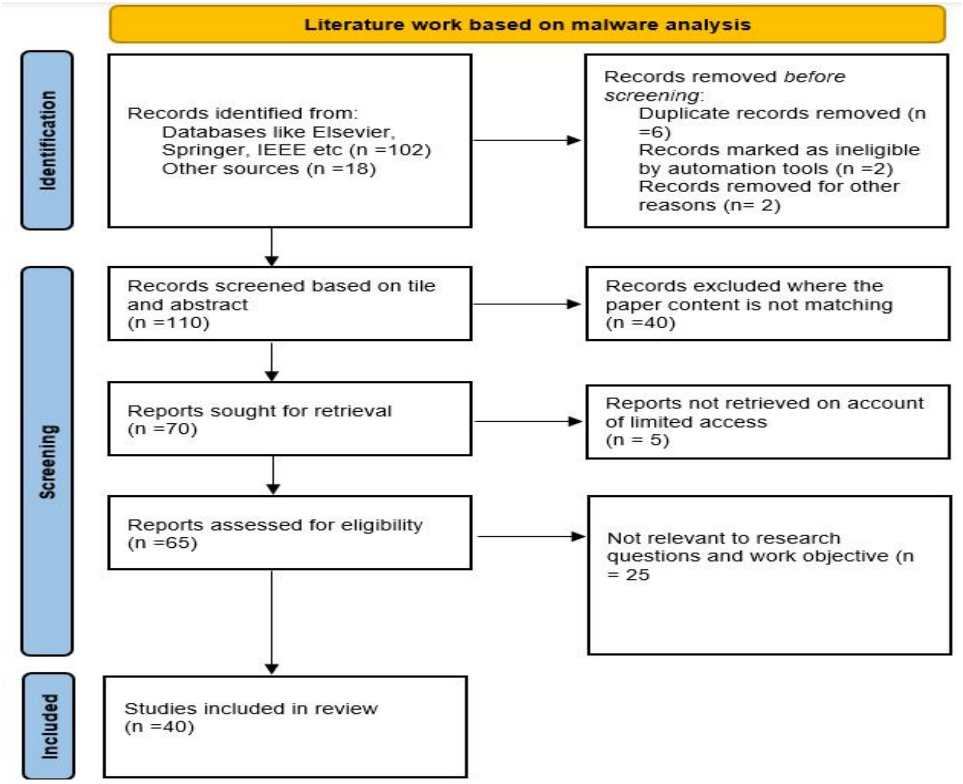

Literature work based on malware analysis
Identification
Screening
Included
Records identified from:
Databases like Elsevier, Springer, IEEE etc (n =102)
Other sources (n =18)
Records removed before screening:
Duplicate records removed (n =6)
Records marked as ineligible by automation tools (n =2)
Records removed for other reasons (n= 2)
Records screened based on tile and abstract (n =110)
Records excluded where the paper content is not matching (n =40)
Reports sought for retrieval (n =70)
Reports not retrieved on account of limited access (n = 5)
Reports assessed for eligibility (n =65)
Not relevant to research questions and work objective (n = 25
Studies included in review (n =40)

Figure 4: Categorization of reviewed literature work for malware analysis

5. Related Works

Experts have offered a wide range of methodologies, trends, and strategies using ML for malware detection and evaluation. Since the discipline of malware evaluation and ML is constantly evolving, this section examines trends in the area that experts have established and provided in recent years.

a. Feature Selection in Malware Identification

Wang et al.[26] presented an enhanced self-variant evolutionary method to identify suitable characteristics for the improvement of Android malware identification. ML models like Random Forest (RF), Logistic Regression (LR), K-Nearest Neighbors (KNN), and Gaussian Naive Bayes (GNB) were used in the suggested method to divide things into two groups. The GNB classifier achieved a maximum accuracy of 95.5%, whereas the KNN classifier attained 93.3% accuracy. Sahin et al.[27] created an ML-based framework for finding malware on Android. It uses a LR-based feature selection strategy to get rid of irrelevant characteristics and find the most relevant permission features, which makes the model more accurate. With RF, the proposed model achieves its highest accuracy rating of 96.45%. Mahindru and Sangal[28] suggested a malware identification technique that utilized Support Vector Machines (SVM). This study focuses on categorizing apps as either malware or benign. The author used a ranking and subgroup feature selection methodology, attaining an identification accuracy of 98.8%. The achieved accuracy demonstrates that the framework is resilient and efficient. We specifically designed this model for binary categorization.

Authors[29] proposed a multi-level feature affinization framework utilizing Information Gain (IG) method for android malware detection. The authors selected three essential attributes for static evaluation: permissions, API calls, and objectives. The maximum precision achieved with RF is 96.28%. Here, the methodology emphasizes binary classification and does not do tests for multi-class malware, which is a limitation. Shatnawi et al.[30] created an effective ML model for Android malware via the Recursive Feature Elimination (RFE) method for reducing features. They employed the LR model to categorize selected attributes into malware and benign classifications. They performed the malware categorization based on two classes of android functionalities: API calls and permissions. The most efficient classifier for the specified data set is the SVM, with an accuracy rate for detection of 94.36%. Nevertheless, the suggested model demonstrated poor performance for recall and F1 value. Hossain et al.[31] developed a feature optimization method utilizing particle swarm optimization (PSO) for the identification of Android ransomware assaults. Afterwards, the authors employ classifiers based on RF and SVM for categorization, with RF demonstrating higher accuracy towards malware datasets. But they have not examined the model on the latest datasets consisting various forms of evolving malware. The ransomware attack was detected with 81.58% accuracy.

In other work[32], the authors built an Android malware identification technique utilizing an Aquila optimizer for optimal solutions. The authors then utilized the LSTM-SVM architecture to scrutinize malware based on permission attributes. It got 97% accuracy for CIC-AndMal-2017 dataset. Nonetheless, this approach demonstrates inadequate memory and F1-score, undermining the thorough assessment of the model's efficacy. Soundrapandian and Subbiah[33] suggested a lightweight machine learning approach for malware identification. To understand the multivariate behavior of features, it used an evolutionary method for feature selection and the Mahalanobis distance metric to decide whether the features were benign or malignant. This model attains a malware detection rate of 95.69% on the CICMalDroid 2020 dataset. This approach is incapable of real-time detection of disguised malware and requires multiclass malware identification. After that, authors[34] developed hybrid feature optimization strategies that combine backpropagation (BP) with the particle swarm optimization (PSO) method to identify malware using optimal features. Their study evaluates its overall performance utilizing four unique malware datasets. Their system is rapid and scalable, employing a parallel computing framework to improve detection accuracy. Their study, however, failed to demonstrate the model's efficiency with multiclass or unknown malware.
The authors[35] proposed an innovative approach for identifying ransomware that utilizes autonomous feature selection using PSO. Based on feature significance, PSO extracts prominent features from various categories of ransomware characteristic data. The authors then assign appropriate attributes to the five ML classifiers to effectively categorize ransomware attacks. Daniel et al.[36] suggested a malware identification structure for Cyber-Physical Systems (CPSs). This study by authors attained exceptional results with an accuracy rate of 98% in binary classification. El-Ghamry et al.[37] introduced a lightweight methodology for detecting malware in image-based IoT systems. This study used the metaheuristic decision-making algorithms, namely ant colony optimization (ACO) and PSO, to identify the best features. This model uses pictures created from packet capture (PCAP) files of network traffic as its dataset. This study employs the PSO technique to optimize the parameters of the SVM classifier, hence improving the precision of binary malware identification. SVM's kernel function consisted of the quadratic kernel function, which improves accuracy. The 95.56% achieved accuracy suggests that it can further enhance the suggested methodology for multiclass and familial malware identification.

b. ML Oriented Malware Identification

ML considered as a subpart of AI utilized for malware identification and threat forecasting. ML methods use input data to execute tasks such as identification, categorization, and pattern recognition through comprehensive data analysis. We conduct a thorough evaluation of relevant literature. This section discusses efforts to identify malware using ML approaches.

Roy et al.[38] introduced an Android malware recognition model employing a ML method to address the temporal bias identified in prior studies. This study identifies the application's sensitive aspects and then employs an aggregate approach to quantify their occurrences. The SVM classifier predicted malware programs with a restricted feature set, achieving 88.72% accuracy, whereas with no feature reduction, it reached 93.35% accuracy. This effort, however, misses a class-wise taxonomy of malware families to identify emerging hidden malware. Later one authors[39] have investigated the difficulties associated with malware identification across extensive data environments. They formulate a weighted voting approach for determining feature relevance and stacking strategies for feature prioritization using ensemble learning. They conducted static and dynamic evaluations of malware using the Cuckoo Sandbox software. The suggested strategy achieves an accuracy of 99.5% using the weighted voting system. Karanja et al.[40] suggested an IoT malware identification approach using ML and Haralick picture texture properties. This study included converting a binary file, which can be either virus or benign software, into a grayscale picture. They calculated the Gray Level Co-occurrence Matrix (GLCM) to get features through the pictures. They used the five extracted Haralick characteristics to classify malware and benign software using RF, KNN, and NB classifiers. The RF strategy produced a maximum accuracy of 95%, but the KNN method acquired an optimal accuracy of 80%.

Surendran et al.[41] developed an Android malware-detecting framework that utilizes graph-based mechanisms for concise feature representation. Here authors attained 99% accuracy on RF classifier. The suggested model is incapable of detecting emulator-based malicious software, making it useless for identifying hidden malware. D'Angelo et al.[42] proposed a malware method of identification that utilized association rules. The suggested framework used the Cuckoo Sandbox tool to perform real-time malware analysis based on API call patterns. An alignment Association rule mining employs an alignment-based approach for repeating subsequences to analyze the behavior of API requests for classified malware. d methodology achieved an accuracy of 99.03% in malware identification. In another work[43], the authors introduced an advanced malware identification technique for the examination of internet protocol addresses in forensic data analysis. They use the DT for identification, achieving an accuracy of 93.5%.

Birman et al.[44] introduced an SPIREL framework for malware detection, using a Deep Reinforcement Learning (DRL) approach to develop a cost-efficient classification model. The technique autonomously modifies the learning algorithms according to the related costs, such as accurate or inaccurate predictions, execution time, and computing resources. The suggested model's accuracy is around 96%. Nonetheless, the suggested architecture is ineffective against adversarial ML attacks. Musikawan et al.[45] presented a sophisticated DL model for the identification of Android malware, namely AMDI-Droid. The AMDI-Droid works in three steps: first, it takes the predictive outputs inherit from hidden layers and puts them together; second, it uses different subnetworks to get accurate attribute representations from the original dataset; and third, it makes a loss function by putting together the predictive losses of all base classifiers connected to each hidden layer. The approach demonstrates inefficiency in detecting new malware and lacks sufficient accuracy in family classification across varying datasets. Subsequently, the authors[46] devised an algorithm for identifying picture malware utilizing ML and DL approaches. The methodologies utilized in identifying malware comprise ANN, LSTM, and InceptionV3. They found that InceptionV3 attained the highest accuracy of 98.76%.

Naeem et al.[47] developed a memory-volatile-dependent malware recognition system using a deep-layered ensemble that combines CNN's weak classifier with meta-learner frameworks. The created solution is platform-independent and monitors the runtime behavior of active processes to identify concealed malware. This model integrates both local and global data to minimize the model's dimensions. They have converted the program executable code for the Windows and Android versions into grayscale and RGB images. But this model fails to identify sophisticated polymorphic malware utilizing the implemented algorithms. Later authors[48] proposed an unnamed MP4 malware identification solution via machine learning methodologies. This work has presented a bidirectional, efficient feature extraction strategy. Further authors[49] created an image-based virus identification system utilizing Transfer Learning (TL) and ML techniques. The authors employ a hybrid model that combines VVG-16 and ResNet-50 to fetch hybrid feature. They progressively improve the proposed bimodal model using a stacking method to increase accuracy. This study achieved complete accuracy with 25 malware classifications in the Malimg dataset. This model consumes a significant amount of computing time, only evaluates a single dataset, and is not capable of real-time virus detection.

c. DL based Malware Identification

The DL model possesses the ability to independently fetch features. This characteristic has raised the frequency of utilizing DL oriented models for malware identification. The literature employs diverse DL-based models for malware identification.

Verma et al.[50] proposed a malware categorization framework for multiple classes of malware families that utilizes the GLCM feature mining method from images. The proposed methodology employs ensemble learning on the Malimg dataset. This approach achieves an accuracy of 98.58%. This technique remains ineffective in accurately finding new malware types and requires a substantial dataset for assessment. Authors[51] developed a system for image dependent malware categorization that employs both classic and TL methodologies to improve malware identification and reduce detection time. However, this framework is limited in terms of picture size, as it only works with specific dimensions, not arbitrary ones. Dib et al. [52] established a system for the categorization of IoT malware and the attribution of its family. The authors did a full study of malware and how it can be put into families. They used a combination of static malware evaluation features and deep learning techniques to find new types of malwares, such as new IoT Mirai versions linked to the COVID-19 pandemic. The classification achieved 99.78% accuracy. They assess this model using only one dataset, which required comprehensive and varied cases of malware for performance assessment.

Authors[53] suggested a graphical malware detection framework employing black and white, RGB along with Markov images. Then the Gabor algorithm was employed on photographs to create Gabor images, and the VGG technique classified malware groups with an F1-measure of 99.97%. MalGan, an analysis by Moti et al. [54] uses a Generative Adversarial Network (GAN) to identify new malware by creating new samples. The IoT malware dataset yielded

99.96% accuracy. Falana et al.[55] used deep GAN and CNNs to create a visualization-based malware recognition model with ensemble learning. To evaluate the model, the Malimg and Malevis datasets on Windows were used. Kumar and Janet[56] used TL to distinguish malware variants. For comparison, they used a CNN oriented design with VGG and ResNet approaches.

To prevent attacks, another authors[57] developed a process for identifying malwares in self-driving cars. They have utilized CNN for malware identification. The model got an accuracy of 97.5% on the Windows malware dataset and 92.2% on the Linux dataset. Other methodologies could potentially improve this model's results. Shaukat et al.[58] gives a hybrid malware identification method combining CNN and SVM using color images. The method consists of three phases, one of which involves transforming PE files into pictures. Similarly, authors[59] introduced a GAN oriented model for malware identification. The authors then executed virus identification by analyzing the visualization patterns of the resulting pictures. The suggested model had a 100% detection rate, showing its great efficacy and resilience. The extended training duration and reduced training stability limit the suggested work.

Apart from the above studied work some other works[60-65] in line with the malware evaluation has been done by the researchers that can also been considered as reference.

6. Comparative Evaluation

This section evaluates and analyzes the literature work that was highlighted in the previous section, using the factors shown in Tables 1 to 3 to reflect the main inferences drawn from it. The dataset serves as a crucial component in assessing the proposed model for malware evaluation, which will evolve over time. Table 1 assesses the various feature selection-based methods for malware analysis, as well as the datasets and identification methods used in the process. The major datasets that are considered are Drebin, CICAndMal2, Virus Total, Virus Share, and Android-based.

Table 1: Comparative evaluation based on feature selection-based methods

Reference Work	Drebin	CICAndMal2	Virus Total	Virus Share	Android based	LR	RF	SVM	NB	Ensemble	Accuracy	Recall
[26]	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✓
[27]	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
[28]	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓
[29]	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	✓
[30]	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓
[31]	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✓	✓
[32]	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓
[33]	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
[34]	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓
[35]	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗
[36]	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
[37]	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓

Table 1 compares malware analysis feature selection methodologies from other studies. Various studies highlight Drebin, CICAndMal2, Virus Total, and Virus Share as a few datasets and metrics. The statistical study shows that most studies value the metrics of accuracy and recall. This suggests that the literature reflects interest in the models' competence in detecting malware (true positives) and in minimizing false negatives. In contrast to RF and SVM, the researchers diverge considerably in their approaches, not employing ensemble methods and having limited use of NB and DT. The differing methods of feature selection raises further questions, especially regarding the flexibility of the malware detection system. The thorough and complex evaluation criteria point to the importance of building sophisticated malware analysis programs.

Table 2: Comparative evaluation based on ML-based methods

Reference Work	Drebin	CIC And Mal 2	Virus Total	Virus Share	DT	KNN	LR	RF	SVM	NB	CNN	Accuracy	Recall
[38]	✓	✓	✕	✕	✕	✕	✕	✕	✓	✕	✕	✓	✓
[39]	✕	✕	✓	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓
[40]	✕	✕	✕	✕	✕	✕	✕	✓	✕	✕	✕	✕	✓
[41]	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓
[42]	✕	✕	✕	✕	✓	✕	✕	✕	✕	✕	✕	✓	✓
[43]	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓
[44]	✕	✕	✓	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓
[45]	✕	✓	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓
[46]	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓
[47]	✕	✓	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓	✓
[48]	✕	✕	✓	✕	✕	✕	✕	✕	✕	✕	✓	✕	✓
[49]	✕	✕	✕	✕	✕	✕	✕	✓	✕	✕	✕	✓	✓

Important information about ML-based malware detection can be seen in Table 2. The software datasets Drebin and Virus Total are frequently used due to their reliability and broad range of features which help in improving the performance of the models. Researchers frequently apply RF and CNN to different problems in malware detection. However, the approaches taken are lacking in variety of algorithms, primarily KNN, LR, and NB, which are not used. The CICAndMal2 dataset is not frequently used which implies there are problems in identification or availability. The malware detection performance improvement is needed which is evidenced by accuracy and recall metrics.

Table 3: Comparative evaluation based on DL-based methods

Reference Work	Malimg	CIC And Mal 2	Virus Total	Virus Share	CNN	VGG	GAN	LSTM	SVM	MLP	TL	Accuracy	Recall

[50]	✓	✕	✕	✕	✓	✕	✕	✕	✕	✕	✕	✓	✓
[51]	✓	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✓	✓
[52]	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✕
[53]	✓	✕	✕	✕	✕	✕	✕	✕	✕	✕	✕	✓	✕
[54]	✕	✕	✕	✓	✓	✕	✓	✕	✕	✕	✕	✓	✕
[55]	✓	✕	✕	✕	✓	✕	✓	✕	✕	✕	✕	✓	✓
[56]	✓	✕	✕	✕	✓	✓	✕	✕	✕	✕	✕	✓	✕
[57]	✓	✕	✕	✕	✓	✕	✕	✕	✕	✕	✕	✓	✓
[58]	✓	✕	✕	✕	✓	✕	✕	✕	✓	✕	✕	✓	✓
[59]	✓	✕	✕	✕	✓	✕	✓	✕	✕	✕	✕	✓	✕

The varied metrics for malware analysis based on DL across different studies are captured in Table 3. Most studies assess measures of accuracy and recall which are vital for evaluating malware detection systems, especially in terms of identifying malware and minimizing false negatives. Conversely, the techniques MLP and LSTM are used less frequently possibly due to their complexity and narrow range of use. CNN and GAN are widely used in various applications. Moreover, several studies frequently omit tools like CICAndMal2 and Virus Total, either due to limitations in dataset availability or a focus on traditional detection methods.

7. Research Challenges

The primary research deficiencies identified in the literature evaluation concerning malware identification and evaluation are outlined below:

- The research indicates that the majority of current feature selection methods employed for malware analysis exhibit low rates of detection and elevated false alarm rates. As a result, creating a good feature selection framework that increases the number of infections found, improves the accuracy of classification for new malware variants, and reduces the number of useless features in large datasets which is at present a big concern.
- Feature selection techniques majorly dependent on static characteristics derived from malware specimens. But polymorphic viruses are constantly changing and can alter their behavior and attributes over time. As a result, developing a method for feature selection with valid features is still under investigation.
- The malware generator employed sophisticated methods to conceal harmful code from the detection system. Consequently, designing a system for detection capable of efficiently identifying the many features of malicious code remains an ongoing issue.
- Malware may affect many OS and equipment, and ML methods developed on a single platform not effectively transfer to others. More work is required to establish malware identification methodologies that can efficiently safeguard diverse scenarios under various platforms. Additionally, identifying zero-day malware requires significant effort.
- With the rapid rise of malware instances, it is important to create ML models that can quickly work and manage extensive datasets while maintaining detection accuracy.
- One difficulty for researchers is their belief that larger data sets result in greater accuracy and less bias. Although this assumption is basic, it is critical to acknowledge the possible issues associated with huge datasets. Utilizing a dataset of moderate size may reduce the computing resources and time necessary for model training.

- To enhance the training procedure and minimize the time needed for processing massive datasets, researchers might consider employing distributed computing approaches and parallel processing. This involves splitting the workload among various computer resources, like GPUs or numerous PCs, to concurrently train the model.

8. Conclusion and Future Scope

Malware analysis is one of the main aspects of cybersecurity because of the growing complexity and prevalence of malicous software. Understanding malware is important for formulating robust defences. Malware analysis enables the identification and classification of various malware types, thereby improving the detection and prevention of future attacks. ML is useful for malware investigation because it can examine enormous datasets and identify complicated patterns. In this paper, we have summarized recent advances in malware analysis using ML, DL, and feature selection through the available literature. We have conducted a comparative evaluation of previous work on malware analysis through various approaches, using metrics such as accuracy, recall, datasets, and specific techniques. We have inferred that the most frequently examined metrics are accuracy and recall, indicating a significant focus on the model's efficacy in accurately detecting malware. When considering ML-based methods for malware evaluation, researchers widely use Drebin and Virus Total datasets, as well as RF and CNN to a large extent. However, DL-based methods for malware evaluation extensively employ CNN and GAN. We also evaluate the obstacles and limitations associated with the studied work, thereby paving the way for future research. In the future integration of ML with other technologies for the mitigation of malwares will be done. Apart from this analysis on other available datasets will also be carried upon.

References

1. Gorment, N. Z. *et al.* (2023) "Machine learning algorithm for malware detection: Taxonomy, current challenges, and future directions," *IEEE access: practical innovations, open solutions*, 11, pp. 141045–141089.
2. Alzarooni, K. M. A. (2012) "Malware Variant Detection," *Doctoral Dissertation*.
3. Stallings, W. *et al.* (2012) *Computer Security: Principles and Practice*. Upper Saddle River, NJ, USA: Pearson Education.
4. Alam, S. *et al.* (2015) "A framework for metamorphic malware analysis and real-time detection," *Computers & security*, 48, pp. 212–233.
5. Mehtab, A., Shahid, W. B. and Yaqoob, T. (2020) "AdDroid: rule based machine learning framework for android malware analysis," *Mobile Networks and Applications*, 25(1), pp. 180–192.
6. Saracino, A. *et al.* (2018) "Madam: Effec tive and efficient behavior-based android malware detection and prevention," *IEEE Transactions on Dependable and Secure Computing*, 15, pp. 83–97.
7. Ucci, D., Aniello, L. and Baldoni, R. (2019) "Survey of machine learning techniques for malware analysis," *Computers & security*, 81, pp. 123–147.
8. Geluvaraj, B., Satwik, P. and Kumar, T. (2019) "The future of cybersecurity: Major role of artificial intelligence, machine learning, and deep learning in cyberspace," in *International Conference on Computer Networks and Com munication Technologies*. Springer, pp. 739–747.
9. Gibert, D., Mateu, C. and Planes, J. (2020) "The rise of machine learning for de tection and classification of malware: Research developments, trends and challenges," *Journal of Network and Computer Applications*, 153.

10. Muttoo, S. K. and Badhani, S. (2017) “Android malware detection: state of the art,” *International journal of information technology*, 9(1), pp. 111–117.
11. Taleby, M. *et al.* (2017) “A survey on smartphones security: Software vulnerabilities, malware, and attacks,” *International journal of advanced computer science and applications : IJACSA*, 8(10).
12. Alzubaidi, L. *et al.* (2021) “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *Journal of big data*, 8(1), p. 53.
13. Qamar, A., Karim, A. and Chang, V. (2019) “Mobile malware attacks: Review, taxonomy & future directions,” *Future generations computer systems: FGCS*, 97, pp. 887–909.
14. Darem, A. *et al.* (2021) “Visu alization and deep-learning-based malware variant detection using opcode level features,” *Future Generation Computer Systems*, 125, pp. 314–323.
15. Aslan, O. and Yilmaz, A. A. (2021) “A new malware classification framework ¨ based on deep learning algorithms,” *Ieee Access*, 9, pp. 936–987.
16. Ijaz, M., Durad, M. H. and Ismail, M. (2019) “Static and dynamic malware analysis using machine learning,” in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE.
17. Lee, J. *et al.* (2021) “Android malware detection using machine learning with feature selection based on the genetic algorithm,” *Mathematics*, 9(21), p. 2813.
18. Tarar, N., Sharma, S. and Krishna, C. R. (2018) “Analysis and classification of android malware using machine learning algorithms,” in *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*. IEEE.
19. Islam, R. *et al.* (2013) “Classification of malware based on integrated static and dynamic features,” *Journal of network and computer applications*, 36(2), pp. 646–656.
20. Saxe, J. and Berlin, K. (2015) “Deep neural network based malware detection using two dimensional binary program features,” in *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE.
21. Nataraj, L. and Manjunath, B. S. (2016b) “SPAM: Signal processing to analyze malware [applications corner],” *IEEE signal processing magazine*, 33(2), pp. 105–117
22. Han, K. S. *et al.* (2015) “Malware analysis using visualized images and entropy graphs,” *International journal of information security*, 14(1), pp. 1–14
23. Yi, T. *et al.* (2023) “Review on the application of deep learning in network attack detection,” *Journal of network and computer applications*, 212(103580), p. 103580.
24. Leka, C. *et al.* (2022) “A comparative analysis of VirusTotal and desktop antivirus detection capabilities,” in *2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, pp. 1–6.
25. *VirusTotal* (no date) *Virustotal.com*. Available at: https://www.virustotal.com/gui/home/search (Accessed: September 9, 2024).
26. Wang, L. *et al.* (2021) “A new feature selection method based on a self-variant genetic algorithm applied to Android malware detection,” *Symmetry*, 13(7), p. 1290.
27. Şahin, D. Ö. *et al.* (2023) “A novel permission-based Android malware detection system using feature selection based on linear regression,” *Neural computing & applications*, 35(7), pp. 4903–4918.
28. Mahindru, A. and Sangal, A. L. (2021) “FSDroid:- A feature selection technique to detect malware from Android using Machine Learning Techniques: FSDroid,” *Multimedia tools and applications*, 80(9), pp. 13271–13323.

29. Bhat, P. and Dutta, K. (2022) “A multi-tiered feature selection model for android malware detection based on Feature discrimination and Information Gain,” *Journal of King Saud University - Computer and Information Sciences*, 34(10), pp. 9464–9477.
30. Shatnawi, A. S., Yassen, Q. and Yateem, A. (2022) “An android malware detection approach based on static feature analysis using machine learning algorithms,” *Procedia computer science*, 201, pp. 653–658.
31. Hossain, M. S. *et al.* (2022) “Android ransomware detection from traffic analysis using metaheuristic feature selection,” *IEEE access: practical innovations, open solutions*, 10, pp. 128754–128763.
32. Grace, M. and Sughasiny, M. (2022) “Malware detection for Android application using Aquila optimizer and Hybrid LSTM-SVM classifier,” *ICST Transactions on Scalable Information Systems*, p. e1.
33. Duraisamy Soundrapandian, P. and Subbiah, G. (2022) “MULBER: Effective Android malware clustering using evolutionary feature selection and Mahalanobis distance metric,” *Symmetry*, 14(10), p. 2221.
34. Al-Andoli, M. N. *et al.* (2022) “Parallel Deep Learning with a hybrid BP-PSO framework for feature extraction and malware classification,” *Applied soft computing*, 131(109756), p. 109756.
35. Abbasi, M. S. *et al.* (2022) “Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection,” *Applied soft computing*, 121(108744), p. 108744.
36. Daniel, A. *et al.* (2023) “Optimal feature selection for malware detection in cyber physical systems using graph convolutional network,” *Computers and Electrical Engineering*, 108.
37. El-Ghamry, A. *et al.* (2023) “Optimized and efficient image-based IoT malware detection method,” *Electronics*, 12(3), p. 708.
38. Roy, A. *et al.* (2020) “Android malware detection based on vulnerable feature aggregation,” *Procedia computer science*, 173, pp. 345–353.
39. Gupta, D. and Rani, R. (2020) “Improving malware detection using big data and ensemble learning,” *Computers & electrical engineering: an international journal*, 86(106729), p. 106729.
40. Karanja, E. M., Masupe, S. and Jeffrey, M. G. (2020) “Analysis of internet of things malware using image texture features and machine learning techniques,” *Internet of Things*, 9(100153), p. 100153.
41. Surendran, R., Thomas, T. and Emmanuel, S. (2020) “GSDroid: Graph signal based compact feature representation for android malware detection,” *Expert systems with applications*, 159(113581), p. 113581.
42. D’Angelo, G., Ficco, M. and Palmieri, F. (2021) “Association rule-based malware classification using common subsequences of API calls,” *Applied soft computing*, 105(107234), p. 107234.
43. Usman, N. *et al.* (2021) “Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics,” *Future generations computer systems: FGCS*, 118, pp. 124–141.
44. Birman, Y. *et al.* (2022) “Cost-effective ensemble models selection using deep reinforcement learning,” *An international journal on information fusion*, 77, pp. 133–148.
45. Musikawan, P. *et al.* (2023) “An enhanced deep learning neural network for the detection and identification of android malware,” *IEEE internet of things journal*, 10(10), pp. 8560–8577.

46. Ahmed, Mumtaz *et al.* (2023) "An inception V3 approach for malware classification using machine learning and transfer learning," *International Journal of Intelligent Networks*, 4, pp. 11–18.
47. Naeem, H. *et al.* (2023) "Development of a deep stacked ensemble with process based volatile memory forensics for platform independent malware detection and classification," *Expert systems with applications*, 223(119952), p. 119952.
48. Tsafrir, T. *et al.* (2023) "Efficient feature extraction methodologies for unknown MP4-Malware detection using Machine learning algorithms," *Expert systems with applications*, 219(119615), p. 119615.
49. Rustam, F. *et al.* (2023) "Malware detection using image representation of malware data and transfer learning," *Journal of Parallel and Distributed Computing*, 172, pp. 32–50.
50. Verma, V., Muttoo, S. K. and Singh, V. B. (2020) "Multiclass malware classification via first- and second-order texture statistics," *Computers & security*, 97(101895), p. 101895.
51. Sudhakar and Kumar, S. (2021) "MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things," *Future generations computer systems: FGCS*, 125, pp. 334–351.
52. Dib, M. *et al.* (2021) "A multi-dimensional deep learning framework for IoT malware classification and family attribution," *IEEE transactions on network and service management*, 18(2), pp. 1165–1177.
53. Pinhero A M L, A. and Visaggio, C. A. (no date) "Malware detection employed by visualization and deep neural network," *Computer & Security*, 105.
54. Moti, Z. *et al.* (2021) "Generative adversarial network to detect unseen Internet of Things malware," *Ad hoc networks*, 122(102591), p. 102591.
55. Falana, O. J. *et al.* (2022) "Mal-Detect: An intelligent visualization approach for malware detection," *Journal of King Saud University - Computer and Information Sciences*, 34(5), pp. 1968–1983.
56. Kumar, S. and Janet, B. (2022) "DTMIC: Deep transfer learning for malware image classification," *Journal of information security and applications*, 64(103063), p. 103063.
57. Qureshi, A. *et al.* (2022) ""eUF : A framework for detecting overthe-air malicious updates in autonomous vehicles," *Journal of King Saud University - Computer and Information Sciences*, 34, pp. 5456–5467.
58. Shaukat, K., Luo, S. and Varadharajan, V. (2023) "A novel deep learning-based approach for malware detection," *Engineering applications of artificial intelligence*, 122(106030), p. 106030.
59. Saidia Fascí, L. *et al.* (2023) "Disarming visualization-based approaches in malware detection systems," *Computers & security*, 126(103062), p. 103062.
60. Baker del Aguila, R. *et al.* (2024) "Static malware analysis using low-parameter machine learning models," *Computers*, 13(3), p. 59.
61. Kasarapu, S. *et al.* (2024) "Comprehensive analysis of consistency and robustness of machine learning models in malware detection," in *Proceedings of the Great Lakes Symposium on VLSI 2024*. New York, NY, USA: ACM.
62. Bilot, T. *et al.* (2024) "A survey on malware detection with Graph Representation Learning," *ACM computing surveys*, 56(11), pp. 1–36.
63. Qureshi, S. U. *et al.* (2024) "Systematic review of deep learning solutions for malware detection and forensic analysis in IoT," *Journal of King Saud University - Computer and Information Sciences*, 36(8), p. 102164.

64. Thakur, P., Kansal, V. and Rishiwal, V. (2024) “Hybrid deep learning approach based on LSTM and CNN for malware detection,” *Wireless personal communications*, 136(3), pp. 1879–1901.
65. Hossain, G. S. *et al.* (2024) *PDF Malware Detection: Towards Machine Learning Modeling with Explainability Analysis*. IEEE Access.