The Nucleus ***61****, No. 1 (2024) 51-62*

www.thenucleuspak.org.pk

The Nucleus
ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

A Comprehensive Review of Machine Learning-Based Malware Detection Techniques for Windows Platform

Amna Wajid[*1],Tauqir Ahmed[2],Uroosa Bilal Chaudhry[1]

[1]*Department of Computer Science, Rachna College of Engineering and Technology, Gujranwala, Pakistan*

[2]*Department of Computer Science, University of Engineering and Technology, Punjab, Pakistan*

ABSTRACT

The growing threat of windows malware poses an increasing risk to the security of computers and the sensitive information they hold. The exponential rise in malware threats targeting the windows platform necessitates robust and adaptive detection mechanisms. Machine learning (ML) techniques demonstrate effectiveness in identifying windows malware therefore, a thorough analysis of these techniques is essential. This paper presents a comprehensive review of machine learning based techniques which have been proposed by research community for detecting windows malware. The review begins by providing a comparison of this study with the existing reviews. Then, we provide details of different ML based malware detection techniques. These techniques have been assessed on multiple parameters including: dataset used for training and testing, availability of dataset, ML model used for classification, the type of extracted features, analysis type and the metrics employed to measure the effectiveness of technique. Furthermore, the paper highlights the limitations and challenges in this field and suggests potential future research directions. By providing a comprehensive overview and critical analysis of ML-based malware detection techniques proposed for the windows environment, this study aims to guide and inspire further research in handling evolving cyber threats.

Keywords: *Malware Detection, Windows Platform, Machine Learning*

1. Introduction

Nowadays, there is a significant surge in the adoption of windows operating systems (OS). The appeal of windows OS lies in its user-friendly interface, facilitating effortless computer operations for individuals with varied technical expertise. Its widespread popularity is attributed to a multitude of features including hardware independence, compatibility with third-party software, open-source options, entertainment functionalities and various other capabilities. However, the extensive use of windows OS has also attracted malicious actors seeking to exploit its popularity. This heightened interest from attackers has led to the propagation of novel malware types, posing a threat to the confidentiality and integrity of data stored on these systems [1].

Over the past decade, there has been a significant proliferation in the development of computer malware. In the current landscape, cybercriminals use malware as a potent weapon to execute attacks on computer systems. The internet serves as the primary medium for launching malware attacks by utilizing channels such as emails, malicious websites and the distribution of software through drives and downloads. Malicious software encompasses a diverse array including: viruses, Trojan horses, worms, root kits, adware or ransomware [2].

To discern between malware and benign samples, analysts employ static or dynamic analysis techniques. Static analysis, also known as signature-based analysis, identifies malicious behaviors within binary source code segments without executing the applications. On the other hand, dynamic analysis detects malware during the execution of malicious applications by observing their characteristics while running on a host system. Typically conducted in a sandbox environment- a controlled, isolated environment where untrusted software can be run securely and separately from the rest of the system without damaging it- this analysis prevents the actual infection of production systems by malware. However, it is important to note that dynamic analysis consumes more resources and entails higher costs [1].

With the increasing frequency of malware attacks, the significance of deploying reliable classification and detection methods increases. The established realm of computer science, namely machine learning, exhibits significant potential in the realm of windows malware detection. With the capability to discern complex data patterns and acquire knowledge from diverse datasets, algorithms of machine learning prove to be well-suited for the identification of computer malware. The increasing interest in employing ML techniques for malware detection has led to a proliferation of studies in this domain. However, owing to the dispersed nature of existing research in this field, there arises a need for a comprehensive review of ML based techniques for windows malware detection [3].

There exist a lot of comprehensive and systematic review papers for android based malware detection techniques. However, despite the abundance of literature in this area, there remains a notable scarcity in reviews addressing malware detection for the windows platform. Moreover, the existing reviews are predominantly informal and exhibit a limited scope. So, to address this gap, this study aims to present a thorough examination of the current state of the research in windows malware detection through machine learning. The review systematically delves into various machine-learning techniques employed for detecting malware, scrutinizes the metrics which are used for evaluation of performance and presents the limitations and challenges of the currently

[*]Corresponding author: engramnawajid@gmail.com

employed methods. The final section of our review identifies potential avenues for future research in this domain.

2. Literature Review

As the usage of windows OS has become widespread, the corresponding increase in security threats from malware has fueled substantial advancements in machine learning based malware detection in recent times [3]. For providing researchers a review of existing techniques, a lot of review studies exist covering the malware detection techniques for android as well as windows platform.

Ö. A. Aslan and R. Samet [4] have performed a survey on various malware detection techniques. The techniques covered in this survey are heuristic-based, signature-based, behavior-based, cloud-based, model checking-based, mobile devices-based, DL-based and IoT-based techniques. In this review, a summary of malware detection techniques has been presented. Also, the current challenges associated with malware detection and the future recommendations to overcome challenges have been given in this review. The limitation of this review is that it has covered studies till 2019. Also, the survey approach is informal and do not give an in depth review of existing approaches for malware detection.

Q. Wu, X. Zhu and B. Liu [6] have performed a review on ML based methodologies for static android malware detection. In the review, authors have delved into the structural aspects of Android applications, examined a range of static feature resources, assessed machine learning techniques for identifying Android malware, discussed both the merits and constraints associated with these approaches and given future directions for researchers who want to work in this domain. This survey has covered studies up to 2020 and also it has only covered static features based techniques. So, this review could be improved by incorporating dynamic feature extraction techniques and by including recent papers from 2021 to onwards.

E. J. Alqahtani *et. al.* [7] have surveyed ML based malware detection methodologies for android devices. The primary objective of the paper is to survey and summarize the different methods and approaches for detecting Android malware, particularly those that leverage machine learning algorithms.The main limitation of this paper is that, it does not provide in-depth analysis or evaluation of each individual technique. Readers seeking a more detailed understanding of specific methods may need to refer to the original research papers. Moreover, the paper was published in 2019, which means it might lack the incorporation of the latest advancements and techniques in detecting Android malware. The field evolves rapidly, and new approaches or trends may have emerged since the paper's publication.

M. N. U. R. Chowdhury *et. al.* [3] have reviewed the current research trends on malware detection techniques on android platform using ML. Authors have provided an overview of Android malware and the resulting security concerns caused by it. Then, they presented various unsupervised, supervised and DL methodologies that have been used for detection of android malware. But, the data is not presented in tabular form so, it is difficult to analyze the data given in this paper. Moreover, the paper covers malware detection techniques for android platform only.

V. Kouliaridis and G. Kambourakis [8] have conducted a comprehensive survey of cutting-edge techniques for detecting Android malware using machine learning. They achieved this by categorizing and providing concise analyses of the latest research over the past seven years i.e. from 2014 to 2021. Their categorization involved examining the analysis type, feature extraction methods, datasets, machine learning classification techniques and the performance evaluation metrics used in these studies. Furthermore, they offered detailed insights into their findings, emerging research trends, potential challenges and future research directions. It is crucial to note that this survey focuses exclusively on ML techniques for Android malware detection and encompasses research up to the year 2021.

J. Senanayake *et. al.* [9] have conducted a review of the literature with the aim of focusing on the use of ML for malware detection on android mobile devices. The paper categorizes the different machine learning techniques and their effectiveness. The authors have discussed the criteria utilized in assessing the effectiveness of different approaches to detect malware such as: precision, accuracy, recall and F1-measure. Authors have also identified common challenges in Android mobile malware detection and highlighted emerging trends in the field, shedding light on potential areas for future research. The paper provides insights up to 2021. The field of Android malware detection and ML continues to evolve rapidly. Therefore, the review does not include the most recent advancements and techniques developed after 2021. So, this work could be extended further by including studies after 2021.

M. Al-Janabiand A. M. Altamimi [10] have presented a review which covers the malware detection methods using ML along with basic concepts of both topics of malware detection and ML. Various representative research studies were reviewed and classified according to their analysis techniques, whether they employed static, dynamic or hybrid approaches. But the scope of paper is very limited. It has given review of only 10 techniques and covers studies published up to 2019.

In literature, several review papers exist for android malware detection. In studies [11-13], review of ML based android malware detection approaches have been presented. However, we have found only few reviews on windows based malware detection techniques and those existing reviews have a limited scope and cover a few ML based techniques and those reviews are published in 2020 so, they do not cover work done after 2020. A comparison of existing surveys has been performed based on following factors and has been shown in Table 1.

Table1: The Comparison of Existing Surveys

Reference	Year	Survey Approach	Malware Detection Approaches Reviewed		Feature Extraction Type		Malware Detectionfor Platform		Papers Included till
			Machine Learning based	Traditional Approaches	Static	Dynamic	Android	PC	
[4]	2020	Informal	✗	✓	✗	✗	✗	✓	2019
[6]	2021	Formal	✓	✗	✓	✗	✓	✗	2020
[7]	2019	Informal	✓	✗	✓	✗	✓	✗	2018
[3]	2023	Systematic	✓	✗	✓	✓	✓	✗	2022
[8]	2021	Formal	✓	✗	✓	✓	✓	✗	2021
[9]	2021	Systematic	✓	✗	✓	✓	✓	✗	2021
[10]	2020	Formal	✓	✗	✓	✓	✗	✓	2019
This Study	2023	Formal	✓	✗	✓	✓	✗	✓	2023

1. *Reference:* This column indicates the reference number of each study
2. *Year:* This column represents the year in which the study was conducted or published**.**
3. *Survey Approach:* This column specifies the approach used in conducting the survey or review. It can be informal, or systematic.
4. *Malware Detection Approaches Reviewed:*This column indicates whether the study reviewed machine learning-based approaches (✓) or traditional approaches (✗) for malware detection.
5. *Feature Extraction Type:* This column specifies the type of feature extraction techniques used in the study for malware detection.
6. *Malware Detection for Platform:*This column indicates whether the study focused on Android (✓) or PC (✓) platforms for malware detection.
7. *Papers included till:* This column represents the time span of papers included in the selected study.

3. Methodology

3.1 Selection Criteria

A set of criteria has been established to select the studies to incorporate in this survey. These criteria aim to offer a thorough examination of the diverse ML approaches employed for detecting windows malware. Furthermore, the selected criteria will provide a thorough understanding of the present research landscape and its applicability to malware detection. The selection criteria encompass:

1. Relevance: This involves research focused on malware detection on windows OS using ML algorithms.
2. Year of Publication: To stay updated about the latest developments, studies published between 2019 and 2023 have been selected to include in this study.
3. Data Availability: Those studies have been included in this review that have made their dataset accessible to the public research community or have provided sufficient information to enable result reproduction.
4. Methodology: Studies included in this survey, particularly those related to windows malware detection, must employ ML algorithms.
5. Evaluation: The evaluations of ML algorithms in this survey rely on quantitative metrics such as accuracy and robustness.

3.2 Selection of the Studies

An extensive search was conducted across various platforms, including online databases such as IEEE Xplore, Springer, ACM Digital Library, Science Direct and Google Scholar to identify pertinent studies for this study. A collection of keywords associated with windows malware detection and ML guided the search. To search the relevant papers, the following search query was used;

(Artificial Intelligence OR Machine Learning OR Deep Learning) AND (Malware OR Malicious Code OR Virus OR Worms) AND (Detection OR Identification OR Recognition) AND (Technique* OR Method OR Approach)

The preliminary search produced a multiple results, which were subsequently refined based on the selection criteria mentioned in the previous section. To assess the relevance and appropriateness of each study for inclusion in this survey, a meticulous examination of both the abstract and full text was undertaken during this filtering process.

3.3 Data Synthesis and Analysis

The chosen papers underwent a comprehensive scrutiny in the course of data collection and analysis to acquire relevant insights into windows malware detection utilizing machine learning. To guarantee the consistency, comprehensiveness and currency of the findings in this review, the informationwas systematically gathered. The details extracted from each paper are as shown in Fig. 1

Moreover, the information derived from the chosen papers served the purpose of comparing different approaches and identifying potential avenues for future investigation.

Through this analysis, a thorough comprehension of the current state of the field was presented, along with

Ref.	Year	ML Models user for Classifications	Dataset	Dataset Availability	Deep Learning Model used	Feature Extracted from	Best Model	Can detect unknown malware	Analysis Tpye

Fig. 1. Data extracted from different studies

highlighting the primary challenges and opportunities for future research.

4. Overview of Selected Studies and Key Finding

S. Naz, and D. K. Singh [1] have introduced a system employing a static analysis approach to preemptively detect malware before the installation of executable files. Their methodology comprises four primary steps: (1) collecting samples, (2) extracting features, (3) dividing the dataset and (4) classifying executable files. The malware analysis involves comparing the extracted features of the PE file. Feature extraction relies on recent research, leveraging prior knowledge of the PE file header. Various techniques like n-datagram, grayscale, and others are utilized for this purpose. In their proposed model, the authors consider the entirety of the PE file header to evaluate features from the executable files. These features are subsequently employed to construct a classifier, aiding in determining the file's malicious nature.

A. Hussain *et. al*. [14] have presented a machine learning-driven method for identifying malicious software in the windows operating system. The proposed model examines distinct characteristics of the PE header file, which it then contrasts with a trained machine learning model to determine if the file is malicious or benign. The study employs six distinct machine learning algorithms to identify malware in windows executable files. The performance of these algorithms is evaluated based on metrics like precision, recall, accuracy and F1-score. The experimental findings indicate that RF outperforms other algorithms, achieving an impressive accuracy of 99.4%.

R. Damaševičius *et. al*. [15] have introduced a methodology for malware detection based on ensemble classification. It uses a two-step classification method, starting with a stacked ensemble of dense and CNN in the first stage and then employing a meta-learner for the final stage classification. The study explores and compares 14 different classifiers for the meta-learner. For comparison, 13 machine learning techniques are utilized, including kNN, Linear SVM, Radial basis function (RBF) SVM, RF, AdaBoost, DT, ExtraTrees, Linear Discriminant Analysis, , Passive Classifier, Logistic Neural Net , Stochastic Gradient Descent classifier and Ridge Classifier. The experiments have been performed using PE headers (ClaMP) dataset. The optimal performance is achieved with an ensemble consisting of five dense and CNN neural networks, combined with the ExtraTrees classifier as the meta-learner.

M. Almousa *et. al.* [16] have emphasized an approach to ransomware detection using Application Programming Interfaces (APIs) in conjunction with ML techniques. The primary objectives of this study include: (i) gaining insights into the ransomware lifecycle on windows platform, (ii) conducting dynamic analyses of samples of ransomware to capture several features associated with malicious code patterns and (iii) the development and validation of ransomware detection models based on machine learning, utilizing diverse benign and ransomware samples. Data was gathered from publicly available repositories and underwent analysis using sandbox for sampling purposes. The acquired datasets were employed for constructing machine learning models. The analysis resulted in an impressive ransomware detection accuracy of 99.18% on windows platforms, demonstrating the potential for achieving highly accurate ransomware detection by combining API calls and machine learning models.

A. Irshad *et. al.* [17] have proposed a ML based technique for malware detection. In their work, authors have extracted features from JSON reports generated by Cuckoo for Windows executable files. These features consist of sensitive and confidential data repeatedly found in the JSON report. After feature extraction, a genetic algorithm has been used to identify the most significant optimal features. These selected optimal features are then used as input to train classifiers capable of distinguishing between Malware and Benign files. SVM achieved an accuracy of 81.3%, the NB classifier reached an accuracy of 64.7%, and the RF classifier demonstrated an accuracy of 86.8%.

F. O. Catak *et. al.* [18] have presented their work with primary objective of creating a classification approach for distinguishing various malware types based on their behavior. The research began by constructing a novel dataset that captures the API calls made within the windows OS, reflecting the behavior of malicious software. The dataset encompasses a range of malicious malware types, including Backdoor, Adware, Downloader, Spyware, Dropper, Trojan horse, worm and virus. For classification task, the LSTM (Long Short-Term Memory) method, a well-established technique for handling sequential data has been employed. The outcomes achieved by classifier exhibit an impressive accuracy of up to 95% and an F1-score of 0.83, which is highly satisfactory.

X. Huang *et. al.* [19] have introduced a novel approach for malware detection that leverages deep learning techniques. This method integrates malware visualization technology with a CNN and the neural network architecture is based on the VGG16 network. Authors have put forth a hybrid visualization technique for malware which combines insights from both static and dynamic analyses.

D. Rabadi and S. G. Teo [20] have explored a novel approach to extract dynamic features based on API calls by

examining both the API calls and their corresponding list of arguments. By harnessing machine learning algorithms, authors have developed two methods to identify and categorize windows malware samples. The first method treats the entire list of arguments of each API call as a single feature, while the second method treats each argument of each API call individually as a feature. The findings demonstrate that the proposed approach surpasses recent malware detection methods based on API arguments in terms of accuracy, constraints and the amount of API-related information required. Through experiments, a remarkable accuracy rate of over 99.8992% has been achieved, which outperforms the current state-of-the-art approaches.

K. Sethi *et. al.*[21] have introduced an innovative framework for malware analysis, designed to efficiently detect and categorize malware. The proposed approach hinges on the utilization of two distinct feature selection algorithms to extract the most pertinent features. This not only reduces training time but also improves the accuracy of classification and detection. The experimental findings highlight that the Decision Tree classifier, in particular, yields a high level of accuracy when compared to other classification methods.

Ö. Aslan and A. A. Yilmaz [5] have introduced an innovative deep learning architecture designed for the classification of malware variants through a hybrid model. The primary contribution made by this research lies in the introduction of a new hybrid architecture that optimally combines two extensive pre-trained network models. This architecture is structured into four key stages: data gathering, the design of a deep neural network (DNN) framework, the training of the proposed DNN and evaluation of the trained DNN. The results demonstrate that the proposed approach can classify malware with a notably high level of accuracy, surpassing the performance of existing methods in the field. When evaluated on the Malimg dataset, it achieved an accuracy of 97.78%, outperforming the majority of machine learning-based malware detection techniques.

M. S. Akhtar and T. Feng [22] have introduced a protective mechanism that assessed three machine learning algorithm approaches for malware detection, ultimately selecting the most suitable one. The findings revealed that, in terms of detection accuracy, Decision Trees (DT) performed exceptionally well at 99%, followed by CNN at 98.76% and SVM at 96.41% when compared to other classifiers. The authors systematically evaluated and quantified the detection accuracy of a machine learning classifier employing static analysis to extract features based on PE data, contrasting its performance with two alternative machine learning classifiers.

K. Shaukat *et. al.* [23] have proposed a novel approach whichintroduces a hybrid framework that merges deep transfer learning and ML for malware detection. Initially, deep transfer learning is employed to extract comprehensive deep features from the last fully connected layer of the deep learning model. Subsequently, machine learning models serve as the final detector, effectively leveraging the intrinsic connections between input and output.The efficacy of the suggested framework is confirmed through validation on a compact dataset. The performance of different models has been assessed by initially focusing on a single feature and subsequently incorporating all features for malware classification. The findings indicate that the proposed framework outperforms other contemporary techniques in terms of effectiveness.

M. S. Akhtar and T. Feng [24] have constructed an innovative ensemble of deep neural networks by combining CNN and LSTM techniques. The CNN-LSTM method we introduced is specifically designed for advanced malware detection without the need for feature engineering. The proposed CNN-LSTM approach achieves the highest detection accuracy, reaching 99%, surpassing other methods for malware detection.

M. Ahmed *et. al.* [25] have represented malware signatures as 2D images and employ deep learning techniques to characterize these signatures within the BIG15 dataset covering nine classes. The proposed work assesses the performance of diverse ML and DL technologies for malware classification, including Logistic Regression (LR), CNN, Artificial Neural Network (ANN), transfer learning with CNN and LSTM. The transfer learning technique, specifically utilizing InceptionV3, demonstrates notable performance, surpassing models such as LSTM, achieving an accuracy of 98.76% on the testing dataset and 99.6% on the training dataset.

J. Palša *et. al.* [26] have concentrated on training ML models employing XGBoost and extremely randomized trees algorithms on two datasets derived from static and dynamic analyses of authentic benign and malicious samples. Subsequently, a comparative analysis of their success rates was conducted, both against each other and in comparison to additional algorithms, including RF, DT, SVM and naïve Bayespreviously assessed in authors prior study on the similar datasets. The most effective classification models, utilizing XGBoost algorithm, demonstrate remarkable performance metrics, achieving a detection accuracy of 91.9% on the static analysis dataset. Similarly, on the dynamic analysis dataset, the XGBoost models achieve a detection accuracy of 96.4%.

The Study by G. O. Ganfure *et. al*. [27] has been driven by the necessity for improved ransomware detection techniques capable of identifying both known and new ransomware types effectively and efficiently. This research introduces "DeepWare," a model to detect ransomware that merges DL with hardware performance counter (HPC) insights. Unlike previous approaches that analyze all HPC data at a single time point for each process, DeepWare adopts a more streamlined strategy to visualize HPC data using deep learning to efficiently and effectively identify ransomware. Experimental results across various ransomware types show DeepWare achieving a 98.6% recall score, outperforming existing similar approaches like RATAFIA, OC-SVM and EGB models by 84.41%, 60.93%, and 21%, respectively.

U. Zahoora *et. al.* [28] have introduced CSPE-R, a Cost-Sensitive Pareto Ensemble strategy designed for detecting new ransomware threats. Initially, the framework utilizes an unsupervised deep Contractive Auto Encoder (CAE) to transform the complex feature space into a more uniform and fundamental semantic space. To develop robust features, CSPE-R explores various semantic spaces at different levels of intricacy. Diverse base estimators are trained across these derived subspaces to establish critical connections among different ransomware attack families. Subsequently, a unique Pareto Ensemble-based approach is employed to select the most effective estimators, achieving a balance between false positives and false negatives. Ultimately, the decisions from these chosen estimators are combined to enhance detection capabilities against unfamiliar ransomware threats. Experimental findings demonstrate CSPE-R's effectiveness in identifying zero-day ransomware attacks.

P. Tumuluru *et. al.* [29] have employed machine learning-based static malware research systems to identify Windows-based malwares. The proposed methodology comprises several distinct phases. Firstly, it involves gathering a comprehensive dataset encompassing both malware and non-malware files. Subsequently, a curated dataset is created, followed by the generation of a detailed report. The process entails extracting a myriad of characteristics from the data, encompassing diverse features. Employing a genetic algorithm, the methodology undergoes a meticulous feature selection phase. Finally, detection is executed using different machine learning classifiers including KNN, RF, LR, XGBoost to identify and classify malware instances effectively. After analyzing all the results, it was found that RF Classifier and XGBoost Classification works gives more accuracy as compared to KNN and LR.

M. Kumar [30] has introduced a scalable malware detection system utilizing big data and a machine learning framework. The machine learning model, implemented via Apache Spark to facilitate distributed learning, employs locality-sensitive hashing for efficient malware detection, notably decreasing detection time. The implementation and experimental analysis follow a five-stage iterative process. The model proposed in this study demonstrates a remarkable 99.8% accuracy rate. Moreover, compared to models proposed by other researchers, this approach significantly diminishes learning and malware detection duration.

S. S. Alshamrani [31] has proposed a novel PDF malware identification system utilizing ML. The uniqueness of this system lies in its dual inspection of PDF files statistically and dynamically, resulting in heightened accuracy in identifying the document's nature. Operating without signatures, this method holds promise in discerning unfamiliar and zero-day malware. The experiment evaluates five distinct classifier algorithms to determine the most suitable fit. Assessment metrics such as true positive rate (TPR), precision, false positive rate (FPR), false negative rate (FNR) and F1-score are computed for each classifier algorithm to identify the best approach. Comparative analysis is conducted against existing PDF classification systems. Additionally, a malicious attack simulation is executed, concealing the malicious code within the PDF file during parsing by the PDF parser. The proposed technique achieves an F1-score of 0.986 using the RF classifier, surpassing the cutting edge F1-measure of 0.978. Thus, this method exhibits effectiveness in detecting malware embedded within PDF files compared to existing systems.

F. Alhaidari *et. al.* [32] have proposed a system, named Zero-Day Vigilante (ZeVigilante), with aim to identify malware by integrating both static and dynamic analyses. In contrast to previous studies, the proposed approach incorporates substantial datasets encompassing ample samples for both types of analyses. These meticulously processed datasets serve as the foundation for training and testing various ML classifiers, including RF, NN, DT, kNN, NB and SVM. Notably, the Random Forest (RF) classifier achieves the highest accuracy rates, recording 98.21% for static analysis and 98.92% for dynamic analysis.

W. Z. Zakaria *et. al.* [33] have introduced RENTAKA, a machine learning framework specifically designed for the early detection of crypto-ransomware. Extracted features align with different phases within the ransomware lifecycle. The experimental phase involved assessing five commonly used machine learning classifiers: Naïve Bayes, kNN, Support Vector Machines, Random Forest and J48. This research presents a pre-encryption detection framework for crypto-ransomware utilizing a machine learning approach. Results indicate that support vector machines (SVM) exhibited the highest accuracy and TPR, achieving 97.05% accuracy and a TPR of 0.995 based on experiments.

E. V. P. Kalyan *et. al.* [34] have presented a malware detection system based on DL with focus on detection and categorization of harmful software. This study has introduced a highly accurate and efficient malware detection method utilizing convolutional neural networks (CNNs). The system takes binary files as input and distinguishes between harmful and benign ones. Minimal preprocessing is applied to the binaries, and the network is responsible for discovering features during trainingan important deviation from current convolutional neural networks.The CNN algorithm demonstrates higher accuracy and efficiency compared to alternative algorithms. Upon implementing the algorithm, authors have attained a commendable accuracy of 95%.

Q. Abu Al-Haija *et. al.* [35] have presented a novel detection system designed to analyze PDF documents and differentiate between benign and malware-infected PDF files. The system proposed here utilizes the AdaBoost decision tree, optimized with ideal hyper parameters, trained, and assessed on an extensive and contemporary dataset named Evasive-PDFMal2022. The experimental evaluation showcases an efficient PDF detection system, having an impressive 98.84% accuracy within a brief prediction interval of 2.174 seconds. Consequently, this model surpasses other cutting-edge models within the same research domain. Thus, the proposed system proves to be an effective tool for identifying PDF

malware, exhibiting high detection performance while minimizing detection overhead.

N. A. Azeez *et. al*. [36] have proposed an ensemble learning approach for malware detection, employing a stacked ensemble of fully-connected networks and one-dimensional CNNs for initial classification. Subsequently, a machine learning algorithm has been utilized for final-stage classification. In the selection of a meta-learner, authors have scrutinized and compared 15 ML classifiers. Additionally, for comparative purposes, five machine learning algorithms: NB, DT, RF, AdaBoosting and gradient boosting were employed. The experiments were conducted on the windows PE malware dataset. The most promising outcomes were achieved using an ensemble comprising seven NN, along with the ExtraTrees classifier serving as the last-stage classifier.

N. Loi *et. al.* [37] have introduced a malware classification pipeline designed to categorize Windows Portable Executable files (PEs). Upon receiving a PE sample, the pipeline initially determines whether it's malicious or benign. If classified as malicious, the pipeline proceeds to conduct a detailed analysis to identify its threat type, family, and behavioral traits. This pipeline has been evaluated using the EMBER open-source dataset, comprising roughly 1 million PE samples, which were statically analyzed. The obtained malware detection outcomes align with those from other academic studies within the current state of the art. Additionally, a comprehensive classification of malicious samples has also been performed. The models employed in this pipeline yield understandable results, aiding security analysts in comprehending the decisions made by the automated process.

M. Asam *et. al.* [38] have introduced two fresh approaches for malware classification: the Deep Feature Space-based Malware Classification (DFS-MC) and the Deep Boosted Feature Space-based Malware Classification (DBFS-MC). In the DFS-MC framework, deep features are derived from tailored CNN architectures and are given as input to a SVM for malware classification. Conversely, in the DBFS-MC framework, enhanced discrimination capability is achieved by integrating deep feature spaces from two customized CNN architectures to create amplified feature spaces. Furthermore, the identification of unusual malware involves employing the deep boosted feature space with SVM. The efficacy of these frameworks is assessed using the MalImg dataset and the hold-out cross-validation method. The proposed DBFS-MC demonstrates improved performance in correctly classifying intricate malware types by leveraging feature boosting created through tailored CNNs. Notably, the DBFS-MC classification framework exhibits favorable results in F-score (0.96), accuracy (98.61%), recall (0.96) and precision (0.96) when tested rigorously using 40% previously unseen data.

Conventionally, anti malware solutions rely on signatures to detect known malware. However, this method encounters limitations in identifying obfuscated and packed malware effectively. Recognizing that understanding a program's structural aspects, such as mnemonics, instruction opcodes and API calls, often reveals the root of an issue, M. Ashik *et. al.* [39] have explored the relevance of these features in distinguishing between unpacked malicious and benign executables in their work. Notably, significant features are gathered using Minimum Redundancy and Maximum Relevance (mRMR) techniques and Analysis of Variance (ANOVA). The study uses four datasets for experimentation employing ML and DL methods, including SVM, NB, J48, RF and XGBoost. Additionally, the performance evaluation involves an assortment of DNN, such as Deep Dense networks, CNN-LSTM and 1D-CNN to classify unknown samples, demonstrating promising outcomes particularly with system and API calls. The combination of system/API calls with static features marginally improves performance compared to models solely trained on dynamic features. Furthermore, to enhance accuracy, distinct deep learning methods have been implemented, showcasing a fine-tuned DNN that yields an F1-measure of 99.1% on dataset-2 and F1-measure of 98.48% on Dataset-3.

G. Ahn *et. al.* [40] have employed a machine learning algorithm to attain a detection accuracy exceeding 99% for identifying malicious files in their work. Additionally, a technique for visualizing data using the dynamic-analysis-based MITRE ATT&CK framework have been introduced by authors for malicious file detection. The PE malware dataset underwent classification utilizing Random Forest, Adaboost and Gradient Boosting models. These models demonstrated accuracies of 99.3%, 98.4% and 98.8%, respectively. The analysis of malicious file behavior was derived by visualizing the data through the application of the MITRE ATT&CK matrix .

S. Aurangzeb *et. al.* [41] have demonstrated the value of utilizing a hardware execution profile to unveil the true execution landscape, aiding in the identification of obfuscated ransomware. Authors have assessed the efficacy of features extracted from hardware performance counters in categorizing malignant applications into ransomware and non-ransomware groups, employing various ML algorithms like RF, DT, GB and Extreme GB. The dataset used consists of 80 ransomware applications and 80 non-ransomware applications sourced from the VirusShare platform. The outcomes has highlighted the significant role of extracted hardware features in effectively identifying and detecting ransomware, achieving an F1-score of 0.97 with RF and Extreme GB.

J. Hemalatha *et. al.* [42] have employed a visualization-based technique, representing malware binaries as 2D-images and a DL model for classification. The proposed malware classification system, based on DL, has implemented a reweighted class-balanced loss function within the last classification layer of the DenseNet model. This adaptation significantly enhances performance in classifying malware by addressing imbalanced data concerns. Extensive experiments have been conducted across four standard malware datasets. The results demonstrate that the proposed technique excels in detecting new malwares with heightened accuracy (98.46% for the BIG 2015 dataset, 98.23% for the Malimg dataset, 98.21% for the MaleVis dataset and 89.48% for the unseen

Malicia dataset). Furthermore, it diminishes false-positive occurrences compared to traditional malware detection methods while preserving efficient computational time. Notably, the proposed malware detection method remains effective and reliable against obfuscation attacks.

To tackle the issue of real-time zero-day malware detection, Z. He *et. al.* [43] have introduced an ensemble learning-based technique. This approach aims to enhance the efficacy of conventional malware detectors, even when relying on a constrained set of micro-architectural features obtained in real-time from existing HPCs. Experimental results showcase that the proposed approach, employing AdaBoost ensemble learning on the RF classifier as the primary classifier, achieves impressive results. Specifically, it attains a 92% F-measure and a 95% True Positive Rate (TPR) while maintaining only a 2% false positive rate in detecting zero-day malware, leveraging solely the top 4 micro architectural features.

In Table2, all the techniques have been presented along with important features which have been extracted from original studies.

Table2: ML based Malware Detection Techniques for Windows Platform

Ref.	Year	ML Models used for Classification	Dataset	Dataset Availability	Features Extracted from	Model with best performance	Approach Used	Analysis Type
[1]	2019	SVM,DT, RF, NB classifier	Virusshare and Vxhaven websites	Public	PE Headers	RF with Accuracy: 98.63%	Feature based Approach	Static
[14].	2022	RF, SVM, DT, AdaBoost, GNB,Gradient Boosting	Kaggle and Malware dataset from github	Public	PEHeaders	RF with Accuracy: 99.44%	Feature based Approach	Static
[15]	2022	KNN, SVM, RBF SVM, RF, AdaBoost, DT, ExtraTrees, Linear Discriminant Analysis, Logistic, Neural Net, Passive Classifier, Ridge Classifier and SGD classifier.	ClaMP	Public	PE Headers	ExtraTrees with Accuracy: 98.8%	Feature based Approach	Static
[5]	2021	ResNet-50, AlexNet, CNN	Malimg, Microsoft BIG 2015, Malevis	Public	Windows PE Files	Accuracy: 97.78%	Feature based Approach	Static
[22]	2022	CNN, SVM, DT	dataset provided by the Canadian Institute for Cybersecurity	Private	PE Files	DT with Accuracy: 99%	Feature based Approach	Static
[24]	2022	CNN-LSTM	Kaggle Microsoft	Public	API Calls	Accuracy: 99%	Feature based Approach	Static
[25]	2023	Transfer Learning using Inception-V3	BIG15 dataset by Microsoft	Public	PE Files	Accuracy:98.76 %	Image based Approach	Static
[29]	2022	RF, KNN, XGBoost, LR	Self Collected	Private	Byte Code Files	RF and XGBoost are Better	Feature based Approach	Static
[30]	2022	RF with TLSH	VirusShare, VirusTotal, theZoo, IEEE, Malwr, Lenny Zelter and Contagio malware repositories	Public	PE Files, API Calls	Accuracy: 99.8%	Feature based Approach	Static
[34]	2022	CNN	Self Collected	Private	PE Files	Accuracy: 95%	Image based Approach	Static
[35]	2022	AdaBoost DT	Evasive-PDFMal2022	Public	PDF Files	Accuracy: 98.84%	Feature based Approach	Static
[36]	2021	Ensemble of Dense ANN and 1-D CNN with extra trees	Kaggle	Public	Windows PE Files	Accuracy: 100%	Feature based Approach	Static
[37]	2021	Pipeleine based Gradient Boosting Decison Tree (GBDT)	EMBER	Public	Windows PE Files	TPR: 86.3%	Feature based Approach	Static

[38]	2021	Customized CNN, SVM, ResNet-18 and DenseNet-201	MalImg malware dataset	Available on Request	PE Files	Accuracy: 98.61%	Feature based Approach	Static
[42]	2021	DenseNet	MalImg, BIG 2015, MaleVis, Malicia	Public	PE Binary Files	Accuracy: MalImg: 98.23%, BIG 2015: 98.46%, MaleVis: 98.21%, unseen Malicia: 89.48%	Image based Approach	Static
[27]	2022	Customized CNN	VirusShare	Public	Hardware Performance Counters	Recall: 98.6%	Image based Approach	Dynamic
[28]	2022	Cost-Sensitive Pareto Ensemble classifier, Deep Contractive Autoencoder (DCAE)	Obtained from [44]	Available on Request	Windows API Calls	Recall: 99%	Feature based Approach	Dynamic
[33]	2022	NB, KNN, SVM, RF, J48	Resilient Information System Security (RISS) Research Group	Available on Request	API Calls	SVM with Accuracy: 97.05	Feature based Approach	Dynamic
[41]	2021	RF, DT, GB, Extreme GB	VirusShare	Public	Hardware performance Counter	RF and ExGB with F1-Score: 0.97	Feature based Approach	Dynamic
[43]	2021	Adaboost Ensemble Learning, Random Forest	VirusTotal, VirusShare	Public	Hardware Performance Counter	F1-Score: 92%	Feature based Approach	Dynamic
[32]	2022	RF, Neural Network, KNN, SVM, DT, NB	IEEE DataPort	Public	PE Files, API Calls	RF with Accuracy: 98.92%	Feature based Approach	Both (Static + Dynamic)
[26]	2022	XGBoost, ET	VirusShare, PortableFreeware, PortableApps	Public	PE Files	Accuracy: 91.9% (static), 96.4% (dynamic)	Feature based Approach	Static+Dynamic
[23]	2023	SVM, CNN	Malimg	Public	PE Files	Accuracy: 99.06%	Feature based Approach	Hybrid

5. Evaluation and Discussion

5.1 Overview of the Key Findings

In Table 2, the different ML based techniques for malware detection on windows platform have been presented based on selected factors including ML model used, dataset used, performance of model, Type of files from where features were extracted and approach type.In this section, we will discuss and give an evaluation of these approaches.

Comparison of ML Models:

1. Random Forest (RF) seems to be the best choice across studies, as it is showing highest accuracy in multiple research works (98.63%, 99.44%, 98.8%, 99.18%, 99.37%, 99.8%, 99.06%, 99.8992%, 99%, 99.8%, 99.06%, etc.).
2. Other than RF model, ExtraTrees, XGBoost, SVM, Decision Trees (DT), CNN, AdaBoost have also shown notable accuracy or F1-scores in the range of 97%-99.8% in different datasets.

Dataset Variations:

1. The datasets used by researchers vary from publicly available datasets to self-collected or private datasets.
2. Most of the studies have used publically available dataset for experimentation and performance evaluation of their work.
3. The dataset choice seems to influence the performance of model, with some models giving high accuracy on specific datasets but giving lower accuracy on unseen datasets.

Feature-based vs. Image-based Approaches:

1. Both feature-based and image-based approaches have shown effectiveness in malware detection.
2. Most of the research has been done on feature-based methods and features have been extracted from PE headers, API calls, Windows PE files and hardware performance counters.
3. Only few researchers have worked on Image-based methods using images of API calls or PE files.

Static vs. Dynamic Analysis:

1. Static Analysis examines the code or structure of a file without executing it. It doesn't require execution of file, which makes it faster and less resource-intensive. It can detect known patterns of malware. But it is challenging to detect obfuscated malware using static analysis.
2. Dynamic Analysis involves executing the file in a controlled environment to observe its behavior. It provides valuable insight into the actual behavior of malware, which helps indetection of new or unseen threats. It is slower and more resource-intensive as compared to static analysis.
3. Both methods have their strengths and limitations. The overall effectiveness of malware detection can be enhanced by forming a hybrid technique with integration of static and dynamic analysis techniques.

Private vs. Public Datasets:

1. Studies using private datasets have reported high accuracy, possibly due to customized data collections.
2. Public datasets are commonly used but may not capture the diverse real-world malware, impacting overall adoptability of model.

5.2 Summary of the Key Contributions

From our extensive review of literature on machine learning-based malware detection for windows systems, this review brings forth following key contributions:

Systematic Literature Review:

This paper systematically examines the relevant literature on machine learning methods to detect malware on windows systems. The papers were selected based on specific criteria and a comprehensive analysis was conducted.

An Overview of Machine Learning based Windows Malware Detection Techniques:

The review covers a range of ML algorithms and datasets employed in detecting windows malware. This information can be valuable for researchers and professionals aiming to understand the current state of the art in this area.

Future Directions:

Future directions for ML based malware detection on windows platform has been highlighted in this review. The review proposes ideas to enhance current methods and develop more efficient approaches for this purpose.

By presenting a thorough evaluation of the current state of the research, assessing strengths and weaknesses of existing methods and outlining problems for future research, this review significantly contributes to the domain of ML based windows malware detection. The findings of this paper can guide researchers to develop more effective and efficient detection systems and to contribute towards advancement of future research in this domain.

6. Future Research Directions

- The growing adoption of ML based techniques for malware detection has led to a lot of research in this area. The research community has proposed different techniques to detect malware at an early stage with an effective accuracy. However, there are still some problems which need to be addressed in order to improve detection accuracy. The following are some future research directions for windows malware detection based on the findings of this review:
- The research can be done to study the impact of using a hybrid ML model formed by integrating different models. The use of hybrid model can improve the classification performance.
- The overall effectiveness of malware detection can be enhanced by forming a hybrid approach with integration of static and dynamic analysis.
- There is a need to develop a generalized model which could be able to detect different types of malware effectively and can also adapt to detect new, unseen malwares.
- The work can be done to develop a real time detection system to detect malware as they emerge in real time.
- There is a need to work on expanding the existing datasets and making them more diverse by adding different malware characteristics and behaviors.
- Most of the research is based on windows API calls and PE files which are not inherently security-focused. Future research in windows malware detection should delve deeper into security-oriented attributes like permission requests and system logs.

In summary, there is considerable space for more research in the domain of ML-based windows malware detection. The future research directions from this review aim to advance the field forward, improve the efficiency of windows malware detection techniques and offer a valuable starting point for future studies.

7. Conclusion

The widespread use of windows has drawn the attention of malicious actors looking to take advantage of its popularity. Windows malware poses a significant risk to the security of windows platform and its users. Therefore, detection of windows malware has become a crucial research domain. Several Machine learning based solutions have been presented and implemented by the research community to solve this critical problem. In this paper, we have performed a comprehensive literature review to explore the different ML based techniques to detect the windows malware. Our aim was to provide an in-depth understanding of the current state of the research in this domain, highlight key findings from recent related studies and its limitations and suggesting potential future research problems which still need to be addressed in order to detect malware at an early stage efficiently.

In conclusion, this paper presents a comprehensive review of the current landscape of windows malware detection using ML. The significant findings and contributions outlined in this survey offer valuable insights to researchers. Furthermore, by outlining limitations and suggesting future research directions, this study provides a roadmap for future studies in this field. We consider that this paper will serve as a valuable resource for the researchers working in this area.

Support Vector Machine:	SVM
Decision Tree:	DT
Logistic Regression:	LR
Random Forest:	RF
Naive Bays:	NB
Gaussian Naive Bayes:	GNB
Radial Basis Function:	RBF
Portable Executeable:	PE
Stochastic Gradient Descent:	SGD
Extreme Gradient Boosting:	XGBoost
Extreme Random Trees:	ET
Stochastic Gradient Boosting:	SGB

References

[1] S. Naz and D.K. Singh, "Review of machine learningmethods for windows malware detection," in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6, 2019.

[2] J. Singh and J. Singh, "A survey on machine learning-based malware detection in executable files," Journal of Systems Architecture, vol. 112, pp. 101861, 2021.

[3] M.N.U.R. Chowdhury, A. Haque, H. Soliman, M.S. Hossen, T. Fatima, and Ahmed, I., "Android malware Detection using Machine learning: A Review," arXiv preprint arXiv:2307.02412, 2023.

[4] Ö. Aslan and R. Samet, "A comprehensive review on malware detection approaches," IEEE Access, vol. 8, pp. 6249-6271, 2020.

[5] Ö. Aslan and A.A. Yilmaz, "A new malware classification framework based on deep learning algorithms," IEEE Access, vol. 9, pp. 87936-87951,2021.

[6] Q. Wu, X. Zhu, and B. Liu, "A survey of android malware static detection technology based on machine learning," Mobile Information Systems, vol. 2021, pp. 1-18, 2021.

[7] E.J. Alqahtani, R. Zagrouba, and A. Almuhaideb, "A survey on android malware detection techniques using machine learning algorithms," in 2019 Sixth International Conference on Software Defined Systems (SDS), pp. 110-117, 2019.

[8] V. Kouliaridis and G. Kambourakis, "A comprehensive survey on machine learning techniques for android malware detection," Information, vol. 12, no. 5, pp. 185, 2021.

[9] J. Senanayake, H. Kalutarage, and M. O. Al-Kadri, "Android mobile malware detection using machine learning: A systematic review," Electronics, vol. 10, no. 13,pp. 1606, 2021.

[10] M. Al-Janabi and A. M. Altamimi, "A comparative analysis of machine learning techniques for classification and detection of malware," in 2020 21st International Arab Conference on Information Technology (ACIT), pp. 1-9, 2020.

[11] J. Qiu, J. Zhang, W. Luo, L. Pan, S. Nepal, and Y. Xiang, "A survey of android malware detection with deep neural models," ACM Computing Surveys (CSUR), vol. 53, no. 6, pp. 1-36, 2020.

[12] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A review of android malware detection approaches based on machine learning," IEEE Access, vol. 8, pp. 124579-124607, 2020.

[13] Z. Wang, Q. Liu, and Y. Chi, "Review of android malware detection based on deep learning," IEEE Access, vol. 8, pp. 181102-181126, 2020.

[14] A. Hussain, M. Asif, M. B. Ahmad, T. Mahmood, and M. A. Raza, "Malware detection using machine learning algorithms for windows platform," in Proceedings of International Conference on Information Technology and Applications: ICITA, Singapore: Springer Nature Singapore, pp. 619-632, 2022.

[15] R. Damaševičius, A. Venčkauskas, J. Toldinas, and Š. Grigaliūnas, "Ensemble-based classification using neural networks and machine learning models for windows pe malware detection," Electronics, vol. 10, no. 4, pp. 485, 2021.

[16] M. Almousa, S. Basavaraju, and M. Anwar, "Api-based ransomware detection using machine learning-based threat detection models," in 2021 18th International Conference on Privacy, Security and Trust (PST), pp. 1-7, 2021.

[17] A. Irshad, R. Maurya, M. K. Dutta, R. Burget, and V. Uher, "Feature optimization for runtime analysis of malware in windows operating system using machine learning approach," in 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), pp. 255-260, 2019.

[18] F. O. Catak, A. F. Yazı, O. Elezaj, and J. Ahmed, "Deep learning based Sequential model for malware analysis using Windows exe API Calls," PeerJ Computer Science, vol. 6, pp. e285, 2020.

[19] X. Huang, L. Ma, W. Yang, and Y. Zhong, "A method for Windows malware detection based on deep learning," Journal of Signal Processing Systems, vol. 93, pp. 265-273, 2021.

[20] D. Rabadi and S. G. Teo, "Advanced Windows methods on malware detection and classification," in Annual Computer Security Applications Conference,pp. 54-68, 2020.

[21] K. Sethi, R. Kumar, L. Sethi, P. Bera, and P. K. Patra, "A novel machine learning based malware detection and classification framework," in 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1-4, 2019.

[22] M. S. Akhtar and T. Feng, "Malware Analysis and Detection Using Machine Learning Algorithms," Symmetry, vol. 14, no. 11, pp. 2304, 2022.

[23] K. Shaukat, S. Luo, and V. Varadharajan, "A novel deep learning-based approach for malware detection," Engineering Applications of Artificial Intelligence, vol. 122, pp. 106030, 2023.

[24] M. S. Akhtar and T. Feng, "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time," Symmetry, vol. 14, no. 11, pp. 2308, 2022.

[25] M. Ahmed, N. Afreen, M. Ahmed, M. Sameer, and J. Ahamed, "An inception V3 approach for malware classification using machine learning and transfer learning," International Journal of Intelligent Networks, vol. 4, pp. 11-18, 2023.

[26] J. Palša, N. Ádám, J. Hurtuk, E. Chovancová, B. Madoš, M. Chovanec, and S. Kocan, "Mlmd—a malware-detecting antivirus tool based on the xgboost machine learning algorithm," Applied Sciences, vol. 12, no. 13, pp. 6672, 2022.

[27] G. O. Ganfure, C. F. Wu, Y. H. Chang, and W. K. Shih, "Deepware: Imaging performance counters with deep learning to detect ransomware," IEEE Transactions on Computers, vol. 72, no. 3, pp. 600-613, 2022.

[28] U. Zahoora, A. Khan, M. Rajarajan, S. H. Khan, M. Asam, and T. Jamal, "Ransomware detection using deep learning based unsupervised feature extraction and a cost-sensitive Pareto Ensemble classifier," Scientific Reports, vol. 12, no. 1, pp. 15647, 2022.

[29] P. Tumuluru, L. R. Burra, M. V. V. Reddy, S. Sudarsa, Y. Sreeraman, and A. L. A. Reddy, "APMWMM: Approach to Probe Malware on Windows Machine using Machine Learning," in 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp. 614-619, 2022.

[30] M. Kumar, "Scalable malware detection system using big data and distributed machine learning approach," Soft Computing, vol. 26, no. 8, pp. 3987-4003, 2022.

[31] S. S. Alshamrani, "Design and Analysis of Machine Learning Based Technique for Malware Identification and Classification of Portable

Document Format Files," Security and Communication Networks, vol. 2022, 2022.

[32] F. Alhaidari, N. A. Shaib, M. Alsafi, H. Alharbi, M. Alawami, R. Aljindan, and R. Zagrouba, "ZeVigilante: detecting zero-day malware using machine learning and sandboxing analysis techniques," Computational Intelligence and Neuroscience, vol. 2022, 2022.

[33] W. Z. Zakaria, M. F. Abdollah, O. Mohd, S. W. M. S. M. Yassin, and A. Ariffin, "RENTAKA: A Novel Machine Learning Framework for Crypto-Ransomware Pre-encryption Detection," International Journal of Advanced Computer Science and Applications, vol. 13, no. 5, pp. 378-385, 2022

[34] E. V. P. Kalyan, A. P. Adarsh, S. S. L. Reddy, and P. Renjith, "Detection of malware using CNN," in 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1-6, 2022.

[35] Q. Abu Al-Haija, A. Odeh, and H. Qattous, "PDF malware detection based on optimizable decision trees," Electronics, vol. 11, no. 19, pp. 3142, 2022.

[36] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE malware detection using ensemble learning," Informatics, vol. 8, no. 1, pp. 10, 2021.

[37] N. Loi, C. Borile, and D. Ucci, "Towards an automated pipeline for detecting and classifying malware through machine learning," arXiv preprint arXiv:2106.05625, 2021.

[38] M. Asam,S. J. Hussain, M. Mohatram, S. H. Khan, T. Jamal, A. Zafar, and U. Zahoora, "Detection of exceptional malware variants using deep boosted feature spaces and machine learning," Applied Sciences, vol. 11, no. 21, pp. 10464, 2021.

[39] M. Ashik, A. Jyothish, S. Anandaram, P. Vinod, F. Mercaldo, F. Martinelli, and A. Santone, "Detection of malicious software by analyzing distinct artifacts using machine learning and deep learning algorithms," Electronics, vol. 10, no. 14, pp. 1694, 2021.

[40] G. Ahn, K. Kim, W. Park, and D. Shin, "Malicious file detection method using machine learning and interworking with MITRE ATT&CK framework," Applied Sciences, vol. 12, no. 21, pp. 10761, 2022.

[41] S. Aurangzeb, R. N. B. Rais, M. Aleem, M. A. Islam, and M. A. Iqbal, "On the classification of Microsoft-Windows ransomware using hardware profile," PeerJ Computer Science, vol. 7, pp. e361, 2021.

[42] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for malware detection," Entropy, vol. 23, no. 3, pp. 344, 2021.

[43] Z. He, T. Miari, H. M. Makrani, M. Aliasgari, H. Homayoun, and H. Sayadi, "When machine learning meets hardware cybersecurity: Delving into accurate zero-day malware detection," in 2021 22nd International Symposium on Quality Electronic Design (ISQED), pp. 85-90, 2021.

[44] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," arXiv preprint arXiv:1609.03020, 2016.