Abdul Sadiq, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 13, Issue 12, December 2023, pp 57-65

RESEARCH ARTICLE OPEN ACCESS

Approaches and Techniques for Malware Analysis - A Review

Abdul Sadiq*, Jhansi Priya S*, Akanksh PN*, Dhruva S Kashyap *
**(Students, Department of Information Science and Engineering, BMS Institute of Technology and Management, Bengaluru-64)*
Email: abdulsadiq2732@gmail.com, jhansipriyas272@gmail.com, akankshnandibevor@gmail.com, dhruva685@gmail.com
Dr. M V Sudhamani**
***(Professor, Department of Information Science and Engineering, BMS Institute of Technology and Management, Bengaluru-64)*
Email: dr.mvs@bmsit.in

ABSTRACT
Malware is one of the critical threats to the electronic devices that are network-connected. In general malware is a malicious code which causes huge damages and harms financially and socially. There have been many traditional techniques and approaches for malware detection and analysis. To improve the detection accuracy by finding them at the earliest, many advanced approaches have been proposed. The developments in cloud computing, machine learning and artificial intelligence, malware detection and analysis methods have gained more importance in current years. The most recent methods, an overview of the latest developments for malware detection and analysis are discussed and presented. Also, briefed about the difficulties and potential paths for malware analysis and detection in the future.

Keywords: Malware, detection, analysis, threat

Date of Submission: 01-12-2023 Date of acceptance: 12-12-2023

I. INTRODUCTION

At present we live in a digital age where everyone has access to digital devices and every individual is digitally connected to the internet. Any unwanted or malicious code which causes harm to digital devices and are built to do it intentionally is known as malware. The number of cybercrimes and financial losses brought on by different cyberattacks has significantly increased. And this is contributing rapidly to the degree of harm or damage done financially to its users and organization every day without any limits. This calls for the need of protecting our systems. There are several types of malware or malwares being described with different notations and names, some of them are virus, bots, ransomwares, trojans, worms, rats, rootkits etc. These malwares are designed in such a way that they cause maximum damage to the victims allowing the intruder to remotely access their devices, execution of remote codes, and stealing highly confidential data or disrupting its access to the user.

Earlier malwares were easier to detect as they had some specific type of signature and had a single process, thus we classify these kinds of malwares into traditional malwares also referred to as simple malwares. These traditional malwares are less sophisticated and are easy to detect. Whereas the next generation malwares are highly sophisticated and use various process and methods to hide itself from detection and pretends to be a benign or legitimate software or code.

A 78% rise over 2020, ransomware impacted 66% of organizations in 2022, according to Sophos's "The State of Ransomware 2022" report [2]. 3,729 complaints concerning ransomware attacks were received by the FBI's Internet Crime Complaint Centre in 2021. A total of $49.2 million in financial losses were caused by the attacks. The 2022 "Verizon Data Breach Investigations Report" state that the ransomware assaults increased significantly in 2022 and accounted for 25% of all the breaches [2]. Business sources indicate that the cost of worldwide cybercrime damages will be over $8 trillion in 2023 alone. Over the next three years, the estimated cost will climb at a rate of 15% annually, and by 2025, it might reach $10.5 trillion [1]. These figures are rising quickly both annually and every ten years. It is projected that ransomware attacks may result in financial damage which can exceed $265 billion by 2031[1].

To safeguard the user and the companies from these attacks and prevent financial and social

losses, malware detection is a very important process. The process of detecting and classifying various malwares into its families, analyzing them based on their signatures and behaviors is generally referred to as malware analysis. The traditional or early malware could be easily detected using signature or heuristic based approaches. But, these approaches have many limitations, as they cannot detect the mutated or self-evolving next generation malwares which are usually unknown and new. With the present world, where rapid development is seen in several domains like Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and so on, has led to the progress of evasive malwares which can trick the detection systems and are highly sophisticated and self-evolving.

Several strategies such as multidisciplinary conceptions of detection that combine ML and DL algorithms for the identification of malware which are anticipated to surface in the future, have been created to combat these new types of malwares. In recent years, numerous techniques utilizing various DL or ML algorithms have been put forth with the aim of detection. It has been found that these techniques are effective in detection with a noticeably high efficiency and accuracy rate. Each approach considers different variables in order to locate, classify, and investigate malware. Each technique has its own benefits and drawbacks, and one may be more effective than the other based on the specifics. There is currently no method that can identify every new generation of sophisticated malware, despite the many strategies that have been proposed.

II. LITERATURE SURVEY

2.1 Static and Dynamic Malware Analysis

Static and dynamic methods are used in malware analysis to evaluate the damage and ascertain the level of sophistication of the intruder. Dynamic analysis monitors the sample in a secure environment, static analysis looks at malware without running it. A thorough grasp of malware behavior and characteristics can be obtained using sophisticated analysis techniques. For simple static analysis, programs like VirusTotal, MD5Deep, PEid, and PEView are utilized. Tools like Dependency Walker and IDA Pro are used for advanced static analysis. For comprehending and evaluating malware, reverse engineering, debugging, disassembly, and methods like packers and obfuscation are crucial. These techniques aid in learning about the malware and recognizing its traits. Reverse engineering's primary goal is to obscure the code's design and make analysis more challenging. Tools that are automated can be used, or it can be completed manually.

[3] In the experiments, the malware sample was run and in-depth analysis was performed using programs like PEiD (Portable Executable iD) detector, PEview, and Wireshark. Designed as a Trojan, malware QQQ.exe was generated and is intended to infect Intel 386 and later processors. The file size is 140kb, and it imports ADVAPI32.dll, KERNEL32.dll, and SHELL32.dll. Malware QQQ.exe communicates distant servers, disables security systems, and uses a lot of RAM (Random Access Memory) once it has infected a system. It functions as ransomware, executing files dropped to %Public% and demanding a ransom on a designated bitcoin address. Modern static and dynamic analysis techniques were used to improve knowledge of the traits and behavior of the malware. These techniques offered comprehensive details about the malware's properties, network connections and system impact.

2.2 Machine Learning-Based Techniques for Malware Analysis

Discussing the use of ML techniques for both static and dynamic analysis [4]. Dynamic malware analysis is done using tools such as Cuckoo sandbox, analyses over 2300 features from malware and achieves 94.64% accuracy, Static analysis on the other hand, achieves 99.36% accuracy but is limited by sophisticated malware behavior. Malware, which includes executables, scripts and downloader codes, fall into several categories such as virus, trojan horse, worm and backdoor. It frequently exhibits complex features that make classification difficult.

PEFILE extracts static features, while dynamic analysis uses Cuckoo sandbox to log registry changes, API calls, and file activities. Previous research investigates Support Vector Machine (SVM)-based detection, function call monitoring, and hybrid approaches integrating static and dynamic methods, including deep neural networks, to analyze malware behavior. Cuckoo sandbox isolates systems for malware execution analysis, offering insights into file changes, registry modifications, API calls and network activities like Domain Name System (DNS) queries and IP accesses. Future directions call for the development of a covert dynamic analysis environment, the use of Deep Neural Networks (DNNs) to enhance classification and the utilization of larger datasets to yield more reliable results. Because malware is obfuscated, it is imperative to improve static analysis in a dynamic environment. Running malware in a dynamic environment facilitates the effective extraction of static features, which enhances malware detection capabilities.

To detect and classify the malware in downloaded files [14] has used several machine learning methods. In the context of malware

detection in downloaded files, a few ML models including supervised and unsupervised learning techniques are investigated. With the best accuracy of 99.99% on the test dataset, the Random Forest Classifier shines out and demonstrates how machine learning can be used to distinguish between files that are harmful and those that are benign. The study tackles the difficulties associated with detecting malware, including the intricacy of conventional detection techniques and the evasive strategies employed by contemporary malware. It explores the trade-offs associated with utilizing machine learning models for malware detection and the complexities of managing data that is skewed. This also covers models and techniques used in malware detection, such as behavior - based recognition, encrypting traffic detection and static detection technology for malware that target Android devices.

Illustrating the need for efficient malware detection in computer system security by offering a perceptive analysis of the threat posed by malware and the shortcomings of current detection techniques [7]. It presents a machine learning-based dynamic method to malware analysis, with a specific emphasis on Windows-based malware detection. Detailed instructions are followed in the process, which starts with gathering datasets and ends with selecting and extracting features before using machine learning classifiers. The performance of three classifiers—SVM, Naive Bayes and Random Forest—in Windows-based malware detection is demonstrated through a summary of key findings. Using 41 selected features, the Random Forest classifier and Genetic Algorithm feature selection together yielded the greatest accuracy of 86.8% among the tested classifiers.

One important stage for successful malware identification is the feature extraction procedure, which uses data from JSON reports such as Dynamic Link Library (DLL) files, registry keys and folders. It highlights how crucial genetic algorithms are for identifying important characteristics in Windows executable files that can be used to identify malware. Highlighting the gains in sensitivity, specificity, and accuracy following feature selection by presenting performance figures for each classifier before and after the feature selection is done. The SVM, naive bayes and random forest classifiers' total performance data are compiled, showing that they have an accuracy of 81.3%, 64.7%, and 86.8% for Windows-based malware detection, respectively. Lastly, it does a good job of illustrating the usefulness and promise of using machine learning methods for Windows-based malware detection, especially Random Forest combined with Genetic Algorithm-based feature selection.

Considering an emphasis on Windows API system function calls, the work [6] offers a thorough examination of behavioral frequency analysis as a powerful method for differentiating between malicious and benign applications. It draws attention to the shortcomings of signature-based defense against unidentified threats and the insufficiency of focusing just on known threats to guarantee system security. Based on Windows API call frequencies, the study assesses the efficacy of different machine learning models in identifying software that is malicious or benign. It employs the use of datasets, applying several algorithms such as Random Forest, K-Nearest Neighbors and Logistic Regression by tracking how well each does in classification. In terms of methodology, it uses several sample datasets and splitting strategies for both validation and training, estimating algorithm accuracy using cross-validation. The outcomes demonstrate that different datasets and models have differing bias and accuracy, with Decision Trees trained by using entropy being more accurate in some circumstances. This also emphasizes how typical Windows API calls found during dynamic malware research can be used to fingerprint malicious activities. Deeper understanding of malware behaviors and patterns is made possible by the research, which emphasizes the significance of API call frequency in differentiating between dangerous and benign software.

2.3 Deep Learning-Based Techniques for Malware Analysis

Present-day detection techniques, which depend on static and dynamic analyses, frequently fail to recognize unfamiliar malware in real-time. Machine learning (ML) techniques are being used more and more for more efficient malware analysis to counter evolving evasion techniques like polymorphism. In malware detection, advanced machine learning—particularly deep learning—eliminates the need for extensive feature engineering and outperforms conventional methods.

A comprehensive presentation of approaches for malware analysis and detection, with an emphasis on applying new techniques, [9] suggested an approach that focuses on using deep learning and machine learning models. To improve classification effectiveness and accuracy, feature extraction methods including looking at sections, byte codes, operational codes and system calls were investigated. The key objective of the study was to assess how well deep learning models performed against the more conventional machine learning techniques. It addressed the usage of diverse feature sets and classifiers, highlighting DNNs' advantages in obtaining higher F1 scores, recall, and precision across a range of feature sets. It also discusses

similar efforts in the realm of malware analysis, going over earlier approaches such as image processing, n-grams, static analysis and semi-supervised learning. It draws attention to the expanding use of deep learning methods in particular, neural networks for the detection and interpretation of unsafe executables. The proposed next step entails applying recurrent neural networks (RNNs) and convolutional neural networks (CNNs) for malware classification, as well as evaluating the combined impact of all feature sets on loss and accuracy. In general, highlights how malware detection is changing and stresses the need to continuously investigate and apply cutting-edge machine learning-based and deep learning-based techniques to counter new threats. It also acknowledges the difficulties and potential directions for this dynamic field.

Using a variety of datasets, the study [5] assesses deep learning and classical machine learning architectures. This suggests cutting-edge image processing methods for effective zero-day malware detection. The ongoing refinement of evasion tactics by malware authors challenges existing detection methods. Systems that rely solely on signatures are vulnerable to zero-day attacks and require deep domain expertise. Though dynamic analysis is resistant to obfuscation, it is not as efficient in real-time as machine learning algorithms, which frequently depend on intricate feature engineering. This research presents a scalable hybrid deep learning framework that allows for real-time, reliable, intelligent detection of zero-day malware. This framework combines deep learning and classical machine learning with self-learning techniques to enable thorough analysis.

The work [16] examines several malware detection topics, with particular emphasis on deep learning (DL) and machine learning (ML) models, methods for converting executable files into images, handling unbalanced datasets, and combining ML and DL for enhanced detection. It suggests a system that combines SVM with improved deep learning models, transfer learning, and malware detection. Experiments with multiple datasets, including VirusShare, Malign, and Microsoft malware datasets, show how effective models like VGG16, ResNet50, InceptionV3, and MobileNet are at detecting malware. There is discussion of methods such as pre-trained model optimization, data augmentation, and normalization. A comparative study demonstrates how accurate and efficient the suggested framework is compared to existing approaches. It discusses model performance, the effect of data augmentation, and the importance of deep learning models in malware detection accuracy, such as VGG16, VGG19, and ResNet50. It demands more investigation into countering hostile attacks, investigating group tactics, and examining IoT and Android apps. To put it simply, it's an extensive investigation of ML/DL techniques, datasets, obstacles, and tactics in malware detection with the goal of creating detection systems that are more effective and scalable.

The threat of malicious software on Windows operating systems is growing and the traditional detection techniques like signature-based and static/dynamic analysis are ineffective against malware that is constantly evolving. The study [12] emphasizes the requirement for intelligent systems that can classify unknown executables in real time. It has been suggested that deep learning techniques, in particular Deep Neural Networks (DNNs), are useful for developing malware detection models. References are made to studies that demonstrate the effectiveness of deep learning, neural networks, and machine learning in classifying malware, with test accuracies ranging from 64.14% to 99.74%. There is discussion of several models, approaches, and strategies that make use of sizable datasets such as the EMBER dataset. For the purpose of classifying malware in Windows environments, features extraction, association mining, and API call sequencing are emphasized. It emphasizes the intricacy of contemporary malware, including fileless malware and obfuscation techniques. In an environment where malware may exist, it emphasizes the value of neural networks and deep learning models that can handle a variety of file formats. The article covers deep learning-based techniques like regularization strategies, activation functions, and various loss functions that are meant to reduce overfitting and boost accuracy during training.

2.4 Advance Techniques for Malware Analysis

In today's information technology landscape, malware's rapid evolution, showcasing sophisticated evasion techniques that undermine computer security.

The study [11] examines the evasive behaviors of Windows malware, with a particular emphasis on the systematic and long-term investigation of 92 evasive strategies employed by the virus to avoid dynamic analysis environments. It provides a thorough framework for x86 binary evasion analysis, examining 45,375 malware samples that were detected between 2010 and 2019. The analysis shows that during the previous ten years, a slight increase of 12% has been observed in the frequency of evasive malware samples, along with notable changes in the evasion strategies used. Certain methods that are specific to malware and not used by genuine software were found. The response

of the security community to the introduction of new evasive tactics was examined, offering factual proof of the community's impact on the techniques' uptake. Contributions of the study include the most comprehensive taxonomy of evasive approaches, an analysis of the evasive behaviors of malware and legitimate programs, and insights into the relationship between malware families and accepted evasive strategies. The limitations and difficulties of accurately identifying evasive strategies are also covered, with a focus on the importance of the dataset and analysis tool offered to comprehend evasive behaviors. Furthermore, because dynamic malware analysis is limited in its ability to identify harmful behavior, the study highlights the necessity for sophisticated detection tools to overcome the complex evasive techniques employed by malware. It emphasizes how evasive behavior affects malware detection, showing that the samples which exhibit more evasive behavior over a threshold are more likely to be flagged as malicious by classifiers. It also emphasizes how malware simulation can be used for system-call analysis, highlighting the need for more study in this unexplored area.

CNN is one of the most promising methods for malware detection. Using runtime behavioral characteristics from Portable Executable (PE) files, a suggested CNN-based Windows malware detector [10] achieves an astounding 97.968% detection accuracy. The study emphasizes how important it is to visualize malware features, try out grayscale image representations, and use deep learning to effectively extract features from unprocessed data. The CNN-based Windows malware detection model is optimized by feature selection techniques like Chi Square, Information Gain, Mutual Information and Relief, which also recommend important API calls and categories for examination. The Behavior-based Feature Extractor, Feature Selector, Image Generator and CNN modules among others, are part of the suggested approach and help with malware detection and classification. The core of the approach is to extract behavioral reports from PE files, arrange them according to the Malware Instruction Set (MIST), and emphasize the significance of dynamic features such as CAT API calls. The effectiveness of the CNN-based model is demonstrated through experiments, which show high recall, accuracy, precision and F-measure, particularly when the Relief Feature Selection Technique is used as a guide. The CNN-based method is more accurate than other approaches, but it takes a little longer to detect changes than other classifiers.

Another unified method utilizing deep learning and visualization [8]. Runtime code analysis is challenged by malware creators' obfuscation techniques. So as to detect system abnormalities and visually alert users to abnormal behavior patterns, this method [8] proposes the use of image-based techniques. Combining human analysis with visual analytics can speed up malware detection. In visual analytics, similarity mining helps identify anomalies using distance measurements. The suggested hybrid model correctly groups obfuscated malware into distinct families by combining deep learning and similarity mining. Model development and experimental validation using a variety of datasets are contributions. The study concludes with a novel hybrid approach for robust malware detection that combines deep learning and visualization and shows a high degree of classification accuracy. With respect to large datasets, the model provides scalability and computational efficiency. The study also describes the novel approaches for obfuscated malware detection, as well as the experimental setup, datasets, and related works in image-based analysis and visualization tools.

In the work [13], It is emphasized how well deep learning handles raw binary bytes and extracts features from images. A 97% accurate model with fusion feature sets is demonstrated, and multi-view feature integration is suggested for dependable detection. Comprehensive analyses of diverse ML and DL frameworks for malware classification are offered. The suggested method combines many data viewpoints in a multi-view feature fusion strategy for reliable virus detection. Additionally, it thoroughly assesses ML methods and DL model architectures for malware classification. Important discoveries highlight the value of behavioral characteristics, the improved performance of some feature sets, such as PE Import, and image representation using CNN architectures. Notwithstanding the progress made, many obstacles still need to be overcome, like choosing the best feature combinations, dealing with feature extraction complexity, and managing the trade-offs between recall and precision.

Exploring the difficulties of categorizing malware that is constantly changing, emphasizes the move toward the use of deep learning, specifically CNNs, for this purpose by converting assembly instructions into pictures that can be classified [15]. By employing Markov transfer matrices and assembly instructions to create a three-channel image, the MCTVD method improves accuracy. It explains about how to classify malware using both static and dynamic features, with an emphasis on deep learning and image-based techniques. MCTVD uses transfer probabilities and opcode sequences to train CNNs for classification on three-channel images, as well as sequence extraction and assembly instructions visualization. Experiments show that

MCTVD outperforms classical approaches, outperforming well-known architectures such as AlexNet and VGG models in terms of accuracy, precision, and training time in a variety of assessment configurations.

It demonstrates how well MCTVD performs in cross-validation setups and how effective it is with small training sets. The three-channel image's ablation experiments demonstrate the superior performance of the combined image by illuminating the contribution of each channel to accuracy. In conclusion, the work offers a method for classifying malware that makes use of deep learning-based and image-based techniques. It highlights the outcomes of the experiments, the efficiency of MCTVD, and the significance of utilizing assembly instructions to produce insightful images for CNN-based classification.

MalAnalyser is a Windows malware detection program [17]. This describes its creation and performance evaluation. The stages of the MalAnalyser workflow include data preprocessing, feature set construction, information collecting, and classification utilizing algorithms such as SVM, kNN, Decision Trees, and Logistic Regression. The feature selection method GLBPSO exhibits iterative particle changes to reach the best detection accuracy. By using preprocessing methods, classification, and frequent subsequence extraction, the program tackles problems such as repetitive API requests and long sequences. By decreasing the size of the feature set for effective detection, GLBPSO helps in feature selection. Additionally, it contrasts the results with and without GA enrichment, demonstrating notable gains in calculation time and detection rates. Experiments show that MalAnalyser is a highly effective ransom ware detector, outperforming other cutting-edge methods in terms of accuracy, precision, recall, and F score. It highlights how crucial GLBPSO and GA are to expanding the scope of malware pattern identification and, eventually, raising the overall detection rates of both known and unknown malware.

In order to differentiate malware attacks from innocuous activity on Windows computers, the article [18] aims to design a framework called API MalDetect that can automatically find distinct and highly significant patterns from raw and lengthy API call sequences. The chosen technique makes use of Local Interpretable Model-agnostic Explanations (LIME) approach for model interpretability and explainability, and Long Short-Term Memory (LSTM) networks for sequence learning and feature extraction. The multifaceted methodology presented for the API MalDetect framework integrates interpretability methods to explain model predictions and deep learning techniques for sequence learning. The framework mainly uses recurrent neural networks (RNNs) of the Long Short-Term Memory (LSTM) kind to train and extract information from the long and raw API call sequences that are collected from Windows systems.

LSTM networks are perfect for processing and analyzing API call sequences, which frequently contain significant temporal information, because they are good at identifying long-term dependencies and patterns in sequential data. The LSTM model's ability to efficiently train and extract features from the API call sequences makes it possible to identify distinctive and pertinent patterns that can differentiate between benign and malicious actions. The suggested framework was tested in an experimental evaluation by binarily classifying executable files on Windows systems as malicious or benign. The studies' outcomes showed how well the API MalDetect framework classified and identified malicious activity from benign activity, indicating that it has potential as a tool for Windows system malware detection and classification.

To solve the difficulties in identifying and categorizing contemporary malware, the study [20] presents the SDIF CNN architecture. Unlike resource-intensive techniques like static and dynamic analysis, the methodology makes use of transfer learning and fine-tuning of pre-trained CNN models. Using end-to-end feature engineering, SDIF CNN effectively detects and classifies malware found in real-world environments. To manage the complexity of current malware, it makes use of deep CNN models and machine learning. The technology recognizes the difficulties in determining the actual executable from a visualized image and transforms malware files into graphics for enhanced detection. Interestingly, it achieves a remarkable 78% accuracy for real-world malware datasets, resisting obfuscation attempts without the need for resource-intensive strategies. The suggested SDIF CNN architecture's performance results show how well it can identify and categorize contemporary malware that is installed on various types of modern devices. With a response time of 471ms and a test accuracy of 55%, the model demonstrates its potential for practical use. This also highlights the comprehensive nature of the suggested methodology and its validation through empirical data, and offers insights into the experimental setup and evaluation criteria used.

The Research [19] introduces a novel methodology for malware detection, combining image-based machine learning models and memory dump analysis. Initially, memory dumps, recognized for providing valuable insights into portable executable structures and behaviors, are collected for debugging purposes. This entails retrieving data

from all processes in physical memory or specific processes using raw extension files. In a departure from conventional practices, it transforms extracted information from memory dumps into RGB images instead of grayscale pixels, enabling the application of image-based classification techniques for malware detection. The unique visual representation enhances interpretability and facilitates the utilization of various data augmentation and image manipulation methods.

It introduces a robust ensemble feature learning approach, incorporating hard voting, soft voting, and weighted average voting. In cases where multiple classifiers yield incorrect predictions, the model with the highest accuracy is discounted, emphasizing precision. Conversely, if one classifier consistently outperforms others, the recommendation is to use that model independently, showcasing flexibility in the decision-making process. The study's outcomes underscore the efficacy of the proposed methodology in malware detection. Encoding memory dumps as RGB images, coupled with techniques for image transformation, data augmentation, and ensemble feature learning, significantly improves the classification performance of machine learning models. The variability in model performance suggests that the suggested methodology has a discernible impact on the accuracy and effectiveness of malware detection. By simplifying the process and focusing on key elements, the refined content maintains the essence of the methodology while eliminating unnecessary details.

In order to establish a more generalized training environment, in the work [21] researchers eliminated specific malware family labels by using a selected dataset with a balanced distribution of benign and malicious classifications. By taking this step, the model is guaranteed to be impartial towards certain malware families and capable of distinguishing between benign and harmful occurrences. The groundwork for further training and assessment was established by meticulous dataset preparation, which placed a strong emphasis on completeness and the lack of missing data. By using a balanced distribution of malware families, the suggested methodology allows the model to identify and categorize different kinds of malware without displaying any biases.

Researchers used SHAP (SHapley Additive exPlanations) values for model explainability, which clarified aspects impacting categorization choices. This method avoids a "black box" system and offers insightful information by ensuring that the model's high accuracy is supported by clear and understandable rationale. The suggested methodology offers a comprehensive strategy for identifying obfuscated malware and includes careful dataset preparation, efficient model training, and explainability-enhancing SHAP values. Experiments utilizing a five-feature dataset reveal that the suggested method is effective at detecting obfuscated malware, with a stunning 99.89% accuracy rate. The robustness of the model in differentiating between benign and malicious occurrences is indicated by high precision and recall metrics. All things considered, the suggested methodology produces a high-quality, explainable model with high accuracy through SHAP values, making it a viable option for identifying malware that has been obfuscated and strengthening cybersecurity defenses.

To tackle the problem of analyzing obfuscated malware over a network, the study [22] proposes to use a hybrid stacked ensemble model called "MalHyStack." The objective is to create a malware detection strategy that works by utilizing feature engineering and sophisticated classification methods. The MalHyStack model is designed to efficiently detect malware that are obfuscated by taking into account an extensive range of attributes obtained from network traffic data. The model combines the outputs of several basic classifiers using a hybrid stacked ensemble technique, gaining access to their combined predictive potential. This makes it possible for MalHyStack to better distinguish between malicious and lawful network activity and to capture the restraints of obfuscated malware patterns. The attributes and traits of the data are vital to the way the suggested model functions. The model's preprocessing unit uses careful feature transformation and dimensionality reduction approaches to minimize the effects of noise and irrelevant data, and effectively extract meaningful features.

The methodology's evaluation shows that it outperforms other classifier models in terms of coefficient score, indicating that it can more accurately classify malware that are obfuscated. Moreover, MalHyStack's computational effectiveness is emphasized, as it requires less time for testing and training than traditional unified models. As this efficiency is attained without sacrificing the model's resilience and accuracy, MalHyStack is a promising strategy with potential applications in the cybersecurity field. It is anticipated that the evaluation metrics' outcomes will show how well the MalHyStack model detects obfuscated malware in network traffic, underscoring the model's potential for real-world use in cybersecurity contexts.

III. CONCLUSION

After reviewing the papers on malware detection and analysis, they have been proven helpful to understand certain methodologies and constraints. Static as well as Dynamic malware analysis methods can be utilised in different conditions based on the information available. Some of the approaches used machine learning-based techniques and analysed the static features by PE file and API calls, registry changes and has achieved an accuracy of 94.64% for dynamic and 99.36% for static analysis, but it is very expensive.

In comparison to the traditional methods, Deep learning techniques have proven to be more efficient hence CNN works faster with higher accuracy. One of the methods used a CNN-based Windows malware detector that analysed a runtime behavioural characteristic from PE files achieved an accuracy of 97.97% in detection. And we could see an improved accuracy of 99.99% for Random Forest Classifier. However, because of the machine learning- based techniques employed, there is a high likelihood of overfitting.

In Memory-Based Analysis, these methods demonstrate the effectiveness of deep learning, neural networks, and machine learning in classifying malware, with test accuracies ranging from 64.14% to 99.74%. We also found a unique combination of image analysis and memory dumps that had given an insight for the PE structures, which provided a better coverage of the two most memory usage. The problem of identifying disguised malware can be tackled by using SHAP values for model explainability and for impactful classification. The use of a five-feature dataset reveal suggested that the method is effective at detecting obfuscated malware, with a stunning 99.89% accuracy rate.

In summary, these papers demonstrate the diversity of approaches available for malware analysis and detection, emphasizing the effectiveness of machine learning, deep learning, and memory-based techniques, each with its strengths and caveats.

REFERENCES

[1] S. Morgan, 2023 Cybersecurity Almanac: 100 Facts, Figures, Predictions, And Statistics. Cybersecurity Ventures, https://cybersecurity ventures.com/cybersecurity-almanac-2023.

[2] S. M. Kerner, Ransomware trends, statistics, and facts in 2023, TechTarget, https://www. techtarget.com/searchsecurity/feature/Ransom ware-trends-statistics-and-facts, 2023.

[3] Saurabh, "Advance Malware Analysis Using Static and Dynamic Methodology," 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India, pp. 1-5, doi: 10.1109/ICACAT.201 8.8933769, 2018.

[4] M. Ijaz, M. H. Durad and M. Ismail, "Static and Dynamic Malware Analysis Using Machine Learning," 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, pp. 687-691, doi: 10.1109/IBCAST.2019.8667136, 2019.

[5] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," in IEEE Access, vol. 7, pp. 46717-46 738, doi:10.1109/ACCESS.2019.2906934, 2019.

[6] A. Walker and S. Sengupta, "Insights into Malware Detection via Behavioral Frequency Analysis Using Machine Learning," MILCOM 2019 - 2019 IEEE Military Communications, Conference (MILCOM), Norfolk, VA, USA, pp. 1-6, doi:10.1109/MILCOM47813.2019.9021034M , 2019.

[7] Irshad, R. Maurya, M. K. Dutta, R. Burget and V. Uher, "Feature Optimization for Run Time Analysis of Malware in Windows Operating System using Machine Learning Approach," 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, pp. 255-260, doi: 10.1109/TSP.2019.87 68808, 2019.

[8] Sitalakshmi Venkatraman, Mamoun Alazab, R. Vinayakumar, A hybrid deep learning image-based analysis for effective malware detection, Journal of Information Security and Applications, Volume 47, Pages 377-389, ISSN 2214-22126, https://doi.org/10.1016/j.jisa.2019 .06 .006,2019.

[9] R. Patil and W. Deng, "Malware Analysis using Machine Learning and Deep Learning techniques," 2020 SoutheastCon, Raleigh, NC, USA, pp. 1-7, doi: 0.1109/SoutheastCon 44009.2020.9368268, 2020.

[10] S. D. S.L and J. C.D, "Windows Malware Detector Using Convolutional Neural Network Based on Visualization Images," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 1057-1069, doi: 10.1109/TETC.2019.2910086,2021.

[11] Nicola Galloro, Mario Polino, Michele Carminati, Andrea Continella, Stefano Zanero, A Systematical and longitudinal study of evasive behaviors in windows

malware, Computers & Security, Volume 113, 102550, ISSN 0167-4048, https://doi.org/10.1 016 /j.cose.2021.102550, 2022.

[12] Usha Divakarla, K Hemant Kumar Reddy, K Chandrasekaran, A Novel Approach towards Windows Malware Detection System Using Deep Neural Networks, Procedia Computer Science, Volume 215, 2022, Pages 148-157, ISSN 1877-0509, https://doi.org/10.1016/j.proc s.2022.12.017, 2022.

[13] Rajasekhar Chaganti, Vinayakumar Ravi, Tuan D. Pham, A multi-view feature fusion approach for effective malware classification using Deep Learning, Journal of Information Security and Applications, Volume 72, 103402, ISSN 2214-2126,https://doi.org/10.1016/j.jisa.2022.1034 0 2, 2023.

[14] Akshit Kamboj, Priyanshu Kumar, Amit Kumar Bairwa, Sandeep Joshi, Detection of malware in downloaded files using various machine learning models, Egyptian Informatics Journal, Volume 24, Issue 1, Pages 81-94, ISSN 1110-8665,https://doi.org/10.1016/j.eij.2022.12 .002, 2023.

[15] Huaxin Deng, Chun Guo, Guowei Shen, Yunhe Cui, Yuan Ping, MCTVD: A malware classification method based on three-channel visualization and deep learning, Computers & Security, Volume 126, 103084, ISSN 0167-4048,https://doi.org/10.1016/j.cose.2022.1030 84 , 2023.

[16] Kamran Shaukat, Suhuai Luo, Vijay Varadharajan, A novel deep learning-based approach for malware detection, Engineering Applications of Artificial Intelligence, Volume 122, 106030, ISSN 0952-1976, https:// doi.org/10.1016/j.engappai.2023.106030, 2023.

[17] Prachi., Namita Dabas, Prabha Sharma, MalAnalyser: An effective and efficient Windows malware detection method based on API call sequences, Expert Systems with Applications, Volume 230, 120756, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2023 .120756, 2023.

[18] Pascal Maniriho, Abdun Naser Mahmood, Mohammad Jabed Morshed Chowdhury, API-MalDetect: Automated malware detection framework for windows based on API calls and deep learning techniques, Journal of Network and Computer Applications, Volume 218, 103704, ISSN 1084-8045, https://doi.org/10.10 16/j.jnca.2023.103704, 2023.

[19] Lalit Kumar Vashishtha, Kakali Chatterjee, Siddhartha Suman Rout, An Ensemble approach for advance malware memory analysis using Image classification techniques, Journal of Information Security and Applications, Volume 77, 2023, 103561, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2023. 103561, 2023.

[20] Sanjeev Kumar, Kajal Panda, SDIF-CNN: Stacking deep image features using fine-tuned convolution neural network models for real-world malware detection and classification, Applied Soft Computing, Volume 146, 110676, ISSN 1568-4946, https://doi.org/10.10 16/j.asoc.2023.110676, 2023.

[21] Mohammed M. Alani, Atefeh Mashatan, Ali Miri, XMal: A lightweight memory-based explainable obfuscated-malware detector, Computers & Security, Volume 133, 103409, ISSN 0167-4048, https://doi.org/10.10 16/j.cose.2023.103409, 2023.

[22] Kowshik Sankar Roy, Tanim Ahmed, Pritom Biswas Udas, Md. Ebtidaul Karim, Sourav Majumdar, MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis, Intelligent Systems with Applications, Volume 20, 200283, ISSN 2667-3053, https://doi.org/10.1016/j.iswa.2023.200283, 2023.