A novel ensemble-based approach for Windows malware detection

Vikas Verma[1], Arun Malik[1], Isha Batra[1], A. S. M. Sanwar Hosen[2]
[1]School of Computer Science and Engineering, Lovely Professional University, Phagwara, India
[2]Department of Artificial Intelligence and Big Data, Woosong University, Daejeon, South Korea

Article Info

Article history:

Received Jan 20, 2024
Revised Jul 10, 2024
Accepted Jul 26, 2024

Keywords:

Clustering
Ensemble
Machine learning
Malware
N-base classifier

ABSTRACT

The exponential growth of internet-connected devices, particularly accelerated by the COVID-19 pandemic, has brought forth a critical global challenge: safeguarding the security of transmitted information. The integrity and functionality of these devices face significant threats from various forms of malware, leading to behavioral distortions. Consequently, a vital aspect of cybersecurity entails accurately identifying and classifying such malware, enabling the implementation of appropriate countermeasures. Existing literature has explored diverse approaches for malware identification, encompassing static and dynamic analysis techniques like signature-based, behavior-based, and heuristic-based methods. However, these approaches face a key issue of inadequately identifying unknown malware variants, often resulting in misclassifications of new strains as benign. To tackle this challenge, this study introduces a novel ensemble-based approach for identifying and classifying malware on Windows platforms, with a specific focus on detecting new and previously unknown variants. The proposed approach leverages multiple machine learning schemes to identify elusive unknown malware that proves challenging for existing methods.

Corresponding Author:

A. S. M. Sanwar Hosen
Department of Artificial Intelligence and Big Data, Woosong University
Daejeon 34606, South Korea
Email: sanwar@wsu.ac.kr

1. INTRODUCTION

The advent of modern computer technology and the Internet has undeniably simplified and streamlined various aspects of life for people. Nowadays, numerous tasks can be accomplished online, ranging from social networking and communication to financial transactions and monitoring human body changes [1]. Unfortunately, these advancements have also enticed cybercriminals to shift their criminal activities to the virtual realm, away from the physical world. According to Cisco's cybersecurity threat trends report, there were over 14,000 reported cybercrimes in 2021, exerting a substantial financial impact on the global economy, costing millions of dollars [2]. One of the primary methods employed to execute these cybercrimes is through the use of malware. Malware refers to any software that engages in unauthorized and suspicious activities on the computers of its victims. This malicious software encompasses various forms such as viruses, worms, Trojan horses, rootkits, and ransomware [3].

Recent advancements in information security research aim to develop defense techniques and mechanisms capable of detecting and eliminating unknown malware, thereby enhancing computer security, and alleviating the need for frequent antivirus software updates. Cybercrime encompasses a range of malicious activities, including malware that steals sensitive information, initiates distributed denial-of-service attacks that may cause damage to operational systems [4]. These diverse forms of malware have the potential to extract confidential data from businesses. As techniques for crafting and producing malware evolve, the number of grey-listed malware also increases. Similar type of detection and application enables using encryption is also designed using videos collected through camera surveillance [5]. Consequently, there is a critical need for intelligent approaches that enable automatic malware detection. The process of malware detection involves three key steps: i) utilizing appropriate tools to analyze malware, ii) extracting static and features that are dynamic from the analyzed data, and iii) employing malware identification techniques to group relevant features together and distinguish between benign and malicious malware.

Traditional dynamic malware analysis techniques often fall short as they allow malware execution within controlled environments like virtual boxes. In-depth investigations are conducted to understand the execution of the malware, its persistence mechanisms, and methods of spreading, as well as the harm it causes to connected networks and systems [6]. This necessitates a controlled execution environment and solid subject knowledge. Through static analysis techniques, portable executable (PE) files are disassembled fully, and their hexadecimal codes are examined to comprehend the behavior and effects for the malware. Proficiency in assembly code, along with a comprehensive understanding of the malware and its functioning, is crucial for this process, which also requires time and memory resources. Despite these techniques, effectively addressing new viruses is becoming increasingly challenging [7]. Existing literature already offers various approaches for malware identification and classification.

The initial step in malware analysis involves conducting both static and dynamic analyses. Static analysis evaluates or disassembles the logic of the code without executing it, extracting features like application programming interface (API) sequences, opcodes, system calls, and other relevant information. Dynamic analysis, on the other hand, involves executing the malicious code within a secure environment to observe its behavior. While static detection methods rely on signatures that are not universally applicable, rendering them ineffective for detecting new threats, dynamic behavior-based approaches increase detection accuracy at the cost of significant overhead. The main purpose of this research work is to figure out the malwares that get newly introduced in the system. In order to achieve this objective, this work makes the following key contributions:

- Introduction of a novel ensemble-based approach that combines the strengths of the best classifiers and clustering techniques for malware identification and classification.
- Evaluation of the proposed approach’s behavior and efficiency for handling unknown malware, providing guidance for using the proposed approach for malware classification on Windows platforms.

Significant research has been conducted on malware analysis and detection using static, dynamic, and machine learning (ML) techniques [8]. Naz and Singh [9] gave a thorough explanation of how ML is applied to Windows malware detection. Malware detection and classification methods can be categorized into five groups: deep learning (DL), model verification, signature, behavior, and heuristics methods. A signature-based approach for malware identification. Signature-based methods, relying on pattern-matching using byte sequences known as signatures, are widely used for malware detection. However, these methods are susceptible to minor changes in malicious code, posing a challenge in identifying modified or previously unknown malware. Obfuscation technologies can evade signature-based techniques, but they require prior knowledge of malware samples [10]. Darshan and Jaidhar [11] proposed a hybrid strategy that combines a linear support vector classification algorithm with the static and dynamic features of PE files in order to precisely identify the unknown virus. The model was trained on a little dataset, which hindered its ability to achieve excellent accuracy.

Javeed *et al.* [12] have developed an intrusion detection systems (IDS) powered by DL that carry the latent to identify intrusions with improved accuracy. An understandable and reliable IDS for industry 5.0 is presented in this study. To improve the intrusion detection procedure in Industry 5.0, the suggested IDS is built by merging a bidirectional-gated recurrent unit (Bi-GRU), bidirectional long short-term memory networks (BiLSTM), and fully connected layers. Then, they used the shapley additive explanations (SHAP) mechanism to evaluate and comprehend the characteristics that most significantly influenced the choice of the suggested cyber-resilient IDS. The two main approaches that were proposed: using the signature and the heuristic rule discovered, we can accurately detect known malware. Alsmadi and Alqudah [13] highlight current methods for identifying and evaluating malicious programmes.

Research by Kim *et al.* [14], the behavior-based approach was highlighted, utilizing dynamic analysis techniques to extract behavioral aspects such as instruction sequences, network activities, and system calls. Lad and Adamuthe [15] categorise the risky codes PE files at the earlier static analysis phase in order to decide what preventive steps should be implemented in later phases. Imran *et al.* [16] proposed a similarity-based technique using hidden Markov models to classify malware based on API call patterns. However, dynamic analysis may face limitations as malware can change its behavior when executed in virtual settings. Mane *et al.* [17] mentioned the features that best describe the provided training data are learned using a deep neural network (DNN). This uses executable files that are portable to teach the DNN the features. This network-based solutions, thus, has low false positive rates and is efficient at detecting both novel and known malware.

A technique was proposed in [18] for identifying assaults on home equipment. The functioning of home internet of things (IoT) devices as well as other observed activities are sequences of user events that are used to represent user behavior in this method [19]. This technique learns event sequences for each circumstance, taking into account that users behave differently based on the environment of the home, such as the time and temperature. This method generates alternative event sequences by deleting some events and learning the commonly observed sequences in order to reduce the influence of events from other users in the house that are included in the monitored sequence. Tariq and Tariq [20] identified that limited resources of internet of medical things (IoMT) devices are preserved using this paper's scalable, hybrid ML system, which successfully detects IoMT ransomware threats. This sophisticated analysis is essential for accurately identifying and eliminating ransomware threats, providing a strong defense for the security of the IoMT ecosystem.

The progress and current developments in malware analysis and detection techniques have been thoroughly studied in [21]. This study, in particular, concentrates on viewpoints that were either mostly overlooked or sparingly examined in earlier surveys. Two examples include examining the utility of each data type based on the analysis methods used and providing a comprehensive taxonomy for malware detection techniques that provides a more detailed presentation of the detection methodologies than behavioral-based, signature-based, and heuristic-based approaches. Idouglid *et al.* [22] presented a unique intrusion detection model with the purpose of protecting industry 4.0 systems from ever-evolving cyber threats. This is accomplished by utilizing ML and DL techniques for dynamic adaptability.

Another similar approach was outlined in [23], an innovative way to designing an intelligent IDS for a smart consumer electronics (CE) network using software-defined networking (SDN)-orchestrated DL. The SDN architecture, which permits static network infrastructure reconfiguration and manages the dispersed architecture of smart CE networks by separating the data planes and control planes, was initially given consideration in this strategy. Sharma and Mishra [19] have provided a theoretical analysis centred on the ensemble approach in addition to several enlightening insights that provide guidance for developing a "excellent and diverse" detector. Amarnath and Gurulakshmanan [24] underline the fact that the various applications include problem diagnostics, recommendation systems, and risk assessment. When it comes to modeling sequential data, gated recurrent units, also known as GRU, which are a variation of recurrent neural networks (RNN), prove to be an extremely useful tool [25]. When it comes to the success of an IDS, training and testing are both extremely important components.

Nguyen and Reddi [26] have explored various ways to enhance anomaly-based IDS by incorporating ML techniques. With the increasing sophistication of cyberattacks, statistical methods alone are insufficient, leading to the adoption of ML and DL techniques in different domains. ML has shown promise in providing strong resistance against attackers, and support vector machine (SVM), K-means, and artificial neural networks (ANN) are prevalent algorithms in IDS research. An example of a dynamic prototype network that is based on sample adaptation is proposed in [27] for the purpose of few-shot malware detection. Initially, a dynamic convolutional neural network is created in order to carry out dynamic feature embedding based on sample adaptation and to extract deeper semantic information from each sample. Ding and Wang [28] proposed that elderly lives could be saved by fall detection systems (FDS) that inform family members or care takers. A RNN model is used to categorize human motions and automatically determine the fall state.

Singh and Singh [29] focuses on optimizing ML parameters to achieve high accuracy binary file classification into harmful and benign files. The depth of the tree, splitting criteria, n-estimators, learning rate, kernel function, k value, loss function, and other critical parameters are evaluated by API calls in order for ML techniques to generate highly accurate malware classifier results. Intelligent malware detection techniques were suggested [30] to identify polymorphic and previously unidentified malware variants. The proposed system used the FP-growth method to derive rules from API sequences and determine the harmfulness of program files. The system focused on Windows executable format files. For the purpose of adaptive offloading, Byun *et al.* [31] suggested a hybrid prediction model that makes use of k-nearest neighbors (KNN) and SVM machine training. Consequently, the sensor information that is very likely to be the cause of the real device problems may be picked and transmitted, which ultimately results in increased offloading performance.

Numerous studies are being conducted to address the growth of dangerous software and examine malware [32]. However, the generalizability of models based on ANNs may not always be guaranteed. Existing approaches primarily focus on identifying malware that is already known in the literature. In the case of newly identified malware, which may be missed by dynamic and ML-based approaches, these methods exhibit lower accuracy in classifying malware into the correct category and often require high execution time for classification. Therefore, it can be concluded from the technical analysis of existing approaches that it is crucial to remember that the success of any detection strategy depends on a number of variables, including the frequency of updates, the quality of the data, the deployment environment, and the sophistication of malware threats. Organizations frequently use many layers of defense, integrating various detection techniques, to ensure thorough malware protection. Continuous research and development are necessary to remain ahead of new threats as the cyber security landscape changes.

The following is the organizational structure of the remaining sections of this work. In section 2, the required preliminary procedures that were utilized in this investigation are presented. These preliminary procedures include specifics on the approaches for data set collecting, preprocessing, and feature extraction. In section 3, we propose an ensemble approach for malware identification and classification. Subsequently, section 4 conducts a thorough evaluation and analysis of the proposed ensemble approach's performance. Finally, concluding thoughts are presented in section 5 of the study.

2. PRELIMINARIES

The process of determining whether a suspicious entity is malware or benign involves several steps, including malware data collection, pre-processing, feature extraction, transformation, selection, and classification. An overview of the methodology employed in this work is presented in Figure 1.

- Data collection: Samples are collected from Windows-based platforms in various forms, such as images, binary files, bytes, and opcodes. For this study, the malware classification dataset provided by Microsoft is utilized, which can be obtained from the source [28]. In order to perform the analysis part, training data and testing data of 80:20 is used.
- Data pre-processing: Unwanted data, such as digitally signed documents, is removed from the collected dataset, focusing on images and files.
- Feature extraction and reduction: Execution traces are logged by analyzing the malware samples. Data mining techniques are employed to extract malware characteristics from these logs. Data mining involves the discovery of patterns and previously unknown values in large databases. During the extraction of malware features, various elements such as strings, byte sequences, opcodes, assembly instructions, system calls, API calls, and a list of dynamic link libraries (DLLs) can be utilized. The feature extraction process employs principal component analysis (PCA) and a random forest (RF) classifier is employed for feature reduction. This step identifies and eliminates irrelevant features from the data.
- Selection and classification: The proposed ensemble approach is used to extract malware features and perform accurate malware classification. It eliminates unwanted features and enhances the accuracy of the classification process.

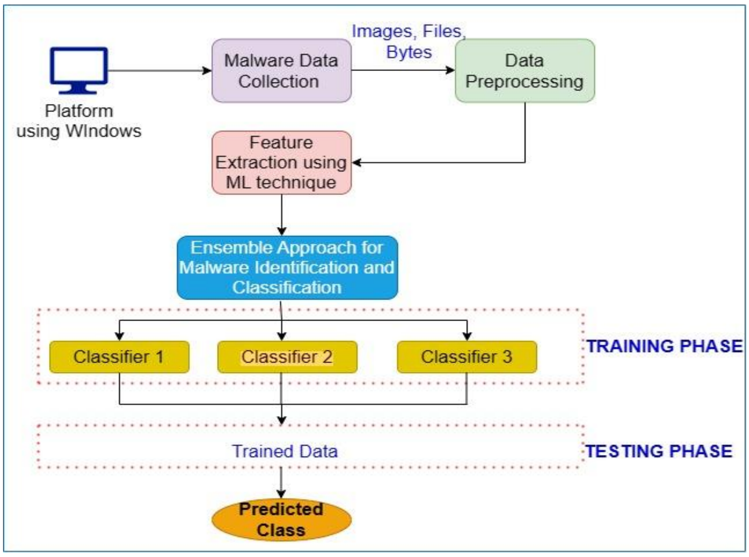

Figure 1. Methodology depicting flow of proposed ensemble approach

3. PROPOSED ENSEMBLE APPROACH FOR MALWARE IDENTIFICATION AND CLASSIFICATION

Detecting and classifying malware poses significant challenges due to the objectives of malware developers, which include information theft, extortion, and network attacks. Traditional methods have been effective in identifying known malware, but they struggle with newly emerged malware, known as zero-day malware. However, the advancement of ML platforms has greatly enhanced the capabilities of malware detection models in identifying threats. ML techniques enable malware detection to be performed in two crucial steps: feature extraction and selection, followed by data classification or clustering. This proposed approach focuses on ML techniques, which can effectively identify both harmful and benign files and accurately predict the nature of previously unseen files.

The proposed approach introduces an ensemble classifier strategy for malware detection and classification. This strategy involves incorporating a base classifier into each modified training dataset, resulting in a collection of base classifiers that form an ensemble. This ensemble formation is the core principle of the approach. To achieve this, the training datasets are reorganized using various resampling or weighting methods, creating multiple variations.

3.1. Ensemble classifier design

It comprises several steps, including the clustering process and the implementation of an ensemble-based classifier for malware identification and classification. The clustering step is conducted prior to applying the ensemble classifier and utilizes the K-means clustering approach to group similar information together. The clustering is based on word frequency, where words with similar frequency indices are clustered into the same group. The number of clusters represented by the centroids is determined based on the desired quantity.

The K-means algorithm begins by selecting initial centers for the clusters from the data patterns at k points. Then, the distance between each sample and the center of its corresponding cluster is calculated, and the sample is assigned to the closest cluster. The average value of the data objects within each newly formed cluster is computed to determine the new center for that cluster. These steps are iteratively repeated until the clustering centers of consecutive iterations do not significantly change, indicating convergence and maximum achievement of the primary clustering function. The ensemble learner approach consists of three phases:

- Phase 1: Preparation of the ensemble involves selecting N base classifiers and choosing a meta-learning algorithm.
- Phase 2: Training of the ensemble occurs by training each of the M base learners using the training dataset. K-fold cross-validation is performed on each base learner, and the predictions are recorded.
- Phase 3: Testing of the ensemble is conducted using new and unknown data. The decisions made by the base learners are recorded, and the meta-learner ensemble decisions are derived from these base-level decisions.

3.2. Selection of n base classifiers for ensemble

The available literature provides a wide range of classifiers, each with its own predictive capabilities. To leverage the strengths of these classifiers and create an innovative ensemble classifier, we adopt the stacked ensemble technique. This approach combines the predictions of diverse base models to achieve improved classification accuracy and reduce the risk of misclassification. In our proposed approach, we incorporate three specific base classifiers:

- SVM: SVMs are a distinctive learning method rooted in statistical learning theory. They are constructed based on a limited number of samples from the training data, aiming to achieve optimal classification results. Initially designed for binary classification tasks, SVMs have been extended to handle large-scale data management and classification in the context of advancements in computer, network, and database technologies.
- Decision tree (DT): DT is a widely employed classification technique with applications in various real-world scenarios. This symbolic learning method constructs a hierarchical structure by analyzing the training dataset. The structure consists of nodes and branches representing different decisions based on the attributes of the dataset.
- Logistic regression (LR): LR is a fundamental statistical and data mining technique widely utilized by statisticians and researchers for analyzing and classifying binary and proportional response datasets. One of its key characteristics is the ability to generate probabilities automatically, making it applicable to both binary and multi-class classification problems.

Various ensemble techniques, including stacking, boosting, blending, and bagging, are available for constructing ensemble models. In this study, we employ the stacking method to create our ensemble. At level 0, SVM and DT models are built, while at level 1, an LR model is constructed. The overall process is illustrated in Figure 2. Once the data has undergone pre-processing, we utilize the term frequency-inverse document frequency (TF-IDF) technique to calculate the frequency of a specific type of malware. The RF model then works on the malware frequency, taking it into account. To generate uncorrelated variables, the data is subjected to PCA, which involves dividing a set of correlated variables into linearly independent subsets. The PCA algorithm processes the malware data with the highest frequency as input and eliminates those with the lowest frequency. This reduces the number of extracted features using the PCA approach. By transforming the data into a lower-dimensional representation, PCA evaluates the effective level of variation present in the data. The PCA technique primarily aims to find a linear transformation vector that maximizes the data variance in the projected space, as represented in (1).

$$t_{k(i)} = w_{l(i)}T_{x_i} \tag{1}$$

Where t is a sequence or vector of values, the subscript $k(i)$ denotes the ith element of a sequence, where k is another sequence or index that specifies the order or position of the elements in t. w is a matrix, where $w_{l(i)}$ represents the i-th row of the matrix. The subscript l(i) refers to the i-th element of the sequence or index l. T_{x_i} denotes the transpose of the vector x_i. x_i represents the i-th input vector. To maximize the variance, the original weight vector w_i must satisfy the following condition, as shown in (2).

$$w_i = (\sum(x_i \cdot w)^2) \tag{2}$$

Where, w_i represents the ith element of the vector w and x_i represents the ith element of the vector x.

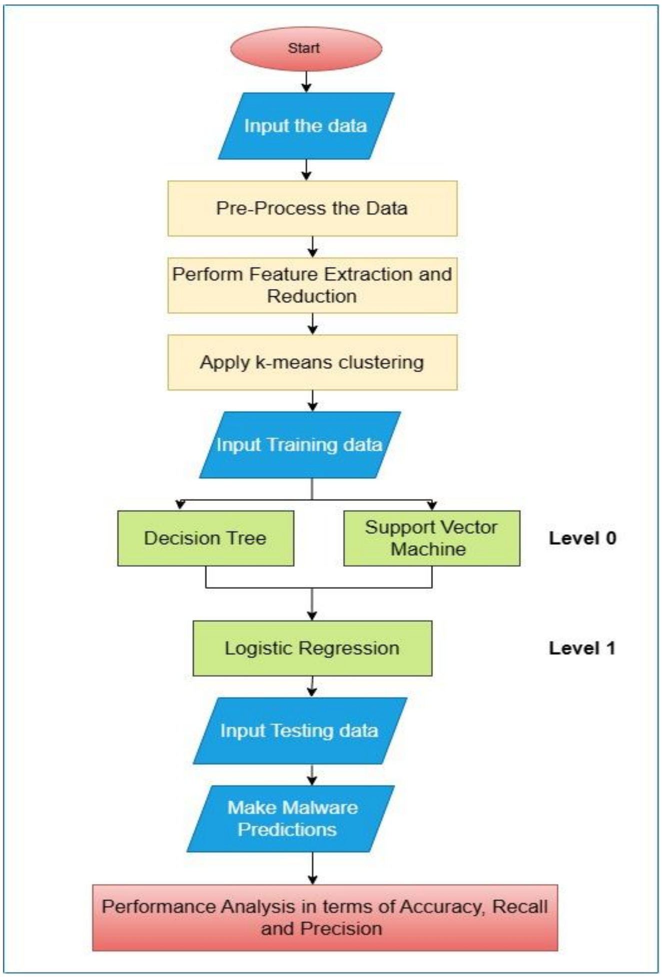

Figure 2. Clustering and classifiers used in Hybrid ensemble approach for malware detection

To group similar information together, an additional clustering step is applied. Malware samples with similar characteristics are clustered together, based on their frequency indices. The number of centroids is equal to the number of clusters, as determined during the calculation. The K-means algorithm starts by selecting k points as the initial cluster centers from the data patterns. Then, the distance between each sample and the center of its corresponding cluster is calculated. The sample is assigned to the closest cluster based on this distance. Afterwards, the average value of each newly formed cluster's data objects is used to calculate the new center for that cluster. These steps are iteratively repeated until the clustering centers of two consecutive iterations do not significantly change. At this point, the clustering process has converged, and the primary clustering objective has been maximized. The algorithm utilizes the Euclidean distance to compute the distance between data samples. The clustering performance is assessed using the sum of squared errors criterion. The K-means technique divides the sample set D= $(x_1, x_2, x_3, \ldots, x_m)$ into $C = (x_1, x_2, x_3, \ldots., x_k)$ clusters for squared error minimization, as shown in (3).

$$E = \sum_{i=1}^{k} \sum x \in \|x - \mu_i\|^2 \quad (3)$$

Where E represents the total sum or cumulative value of the expression on the right-hand side of the equation. It is considered as the result or output of the equation. x represents an individual value or observation in a dataset. In the equation, x is used as a summation variable, indicating that the subsequent expression is evaluated for each value of x. k represents the number of groups or clusters in the dataset. It defines the range or limits of the summation in the equation, specifying that the expression is evaluated for values of i ranging from 1 to k. Here, i represent the index of each group or cluster in the dataset and is used as a summation variable, indicating that the subsequent expression is evaluated for each group or cluster. μ_i represents the mean or centroid of the ith group or cluster. It indicates the average or central value of the observations within that particular group.

Incorporating all the steps, the ensemble approach is developed by combining the DT, SVM, and LR classifiers. DT is used in ensemble as it supports interpretability. When interpretability and transparency are crucial, DTs are a common option since they are easy to comprehend and visualize. SVM is deployed as it has capability to handle high-dimensional data effectively. Moreover, SVMs are very effective for issues when the numbers of features are large as compared to the number of samples. LR generates probability scores between 0 and 1, which represent the possibility of falling into a specific class, rather than binary predictions (0 or 1). Depending on the needs of the application, this probability score may be useful for making judgments, evaluating forecasts, and establishing various decision thresholds.

4. RESULTS AND DISCUSSION

In the field of cybersecurity, malicious actors frequently employ malicious software to carry out cyber-attacks on targeted systems. Malware, which encompasses various forms such as viruses, worms, Trojan horses, rootkits, and ransomware, refers to software designed to intentionally execute harmful actions on unsuspecting victims' computers. Each type and family of malware has its own specific objectives, ranging from compromising system integrity to facilitating the theft of private information and enabling remote code execution.

The study investigated the effects new malware types on windows system and their detection while earlier studies have explored the impact of established techniques. Initially, malware had straightforward objectives, making it relatively easier to detect. This category, known as traditional or basic malware, was identifiable using established techniques. However, the threat landscape has evolved, giving rise to a new generation of kernel-mode malware. These advanced malware variants pose significant challenges in detection compared to older versions. In contrast, conventional malware typically consists of a single procedure and does not employ complex techniques to evade detection.

Furthermore, modern malware utilizes a combination of active and dormant procedures simultaneously, employing various obfuscation techniques to conceal its presence and persist within a network. To address the identification and classification of new or unique categories of malware, numerous ML approaches have been explored in the literature. In this research study, a comparative analysis is conducted among different ML approaches based on metrics such as precision, accuracy, recall, and F1 Score. The analysis utilizes the Microsoft Big dataset [33]. The evaluated ML approaches include SVM, RF, light gradient boosting machine (Light GBM), LR, and DT. Table 1 presents the performance metrics of these ML approaches.

Table 1. Performance analysis of ML approaches for malware classification

ML Approaches	Performance parameters			
	Accuracy	Precision	F1 Score	Recall
SVM	0.95	0.94	0.95	0.95
RF	0.92	0.93	0.93	0.93
Light GBM	0.89	0.88	0.88	0.88
LR	0.96	0.96	0.96	0.95
DT	0.97	0.98	0.97	0.97

During the performance evaluation of different ML approaches for malware classification as depicted in Figures 3 to 6, showcasing the comparison based on precision, accuracy, recall, and F1 score, respectively. We found that that the DT approach outperforms other approaches in identifying and classifying malware across various performance metrics. DT approach outperforms other approaches by getting 98% precision, 97% accuracy, 97% recall, 97% F1 Score. However, even with DT approach, the maximum accuracy achieved is 97%. This emphasizes the necessity for further advancements in developing a more robust system that can enhance the accuracy and effectiveness of malware identification and classification.

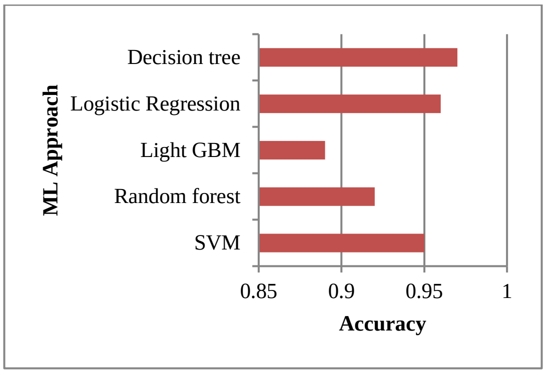

Figure 3. Accuracy comparison of different ML approaches for malware classification

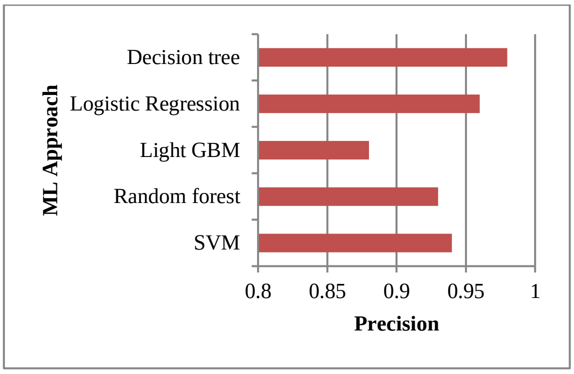

Figure 4. Precision comparison of different ML approaches for malware classification

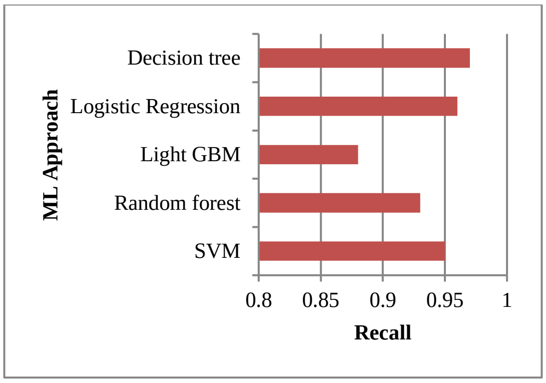

Figure 5. Recall comparison of different ML approaches for malware classification

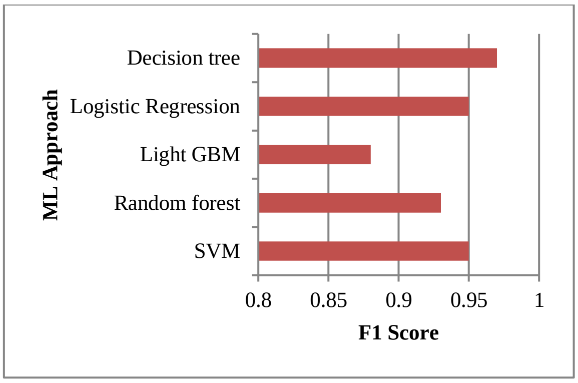

Figure 6. F1 score comparison of different ML approaches for malware classification

5. CONCLUSION

In this study, a comprehensive examination is conducted on various ML techniques available in the existing literature for the identification and classification of malware. Metrics such as accuracy, precision, recall, and F1 score are utilized in order to conduct a comparative analysis of the performance of these various techniques. The results highlight that among the existing variants, RF exhibits superior performance. However, there is still room for enhancement in terms of accuracy, as certain benign files are erroneously labeled as malware and vice versa. To attain optimal outcomes, this research introduces an innovative ensemble-based method for malware identification and classification. The proposed approach integrates multiple ML techniques at different stages, with the objective of addressing and overcoming the challenges associated with incorrect identification and classification. The results prove that DT, LR, and SVM that outperform other existing ML approaches in terms of accuracy, precision, recall and F1 score.

ACKNOWLEDGEMENTS

This work was supported by Woosong University Academic Research Fund in 2024, South Korea.

REFERENCES

[1] K. Kishore and S. Sharma, "Information security & privacy in real life-threats & mitigations: a review," *Journal of Computer Science*, vol. 4, no. 1, pp. 34–67, 2013.

[2] S. Shamneesh, M. Manoj, and K. Keshav, "Node-level self-adaptive network path restructuring technique for internet of things (IoT)," *Advances in Intelligent Systems and Computing*, vol. 989, pp. 453–461, 2020, doi: 10.1007/978-981-13-8618-3_48.

[3] O. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[4] H. Y. Chen, K. Sharma, C. Sharma, and S. Sharma, "Integrating explainable artificial intelligence and blockchain to smart agriculture: research prospects for decision making and improved security," *Smart Agricultural Technology*, vol. 6, 2023, doi: 10.1016/j.atech.2023.100350.

[5] I. Aribilola, M. N. Asghar, N. Kanwal, M. Fleury, and B. Lee, "SecureCam: selective detection and encryption enabled application for dynamic camera surveillance videos," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 2, pp. 156–169, 2023, doi: 10.1109/TCE.2022.3228679.

[6] S. Singh, A. Malik, I. Batra, S. Sharma, and M. Poongodi, "Need for integration of blockchain technology in supply chain management of health supplements," *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2023*, pp. 1757–1761, 2023, doi: 10.1109/ICACITE57410.2023.10183099.

[7] S. O. Subairu *et al.*, "An experimental approach to unravel effects of malware on system network interface," *Advances in Data Sciences, Security and Applications*, vol. 612, pp. 225–235, 2020, doi: 10.1007/978-981-15-0372-6_17.

[8] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, 2018, doi: 10.1186/s13673-018-0125-x.

[9] S. Naz and D. K. Singh, "Review of machine learning methods for windows malware detection," *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, 2019, doi: 10.1109/ICCCNT45670.2019.8944796.

[10] A. A. E. Elhadi, M. A. Maarof, B. I. A. Barry, and H. Hamza, "Enhancing the detection of metamorphic malware using call graphs," *Computers and Security*, vol. 46, pp. 62–78, 2014, doi: 10.1016/j.cose.2014.07.004.

[11] S. L. S. Darshan and C. D. Jaidhar, "Windows malware detection system based on LSVC recommended hybrid features," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 2, pp. 127–146, 2019, doi: 10.1007/s11416-018-0327-9.

[12] D. Javeed, T. Gao, P. Kumar, and A. Jolfaei, "An explainable and resilient intrusion detection system for industry 5.0," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1342–1350, 2024, doi: 10.1109/TCE.2023.3283704.

[13] T. Alsmadi and N. Alqudah, "A Survey on malware detection techniques," *2021 International Conference on Information Technology, ICIT 2021*, pp. 371–376, 2021, doi: 10.1109/ICIT52682.2021.9491765.

[14] H. Kim, J. Kim, Y. Kim, I. Kim, K. J. Kim, and H. Kim, "Improvement of malware detection and classification using API call sequence alignment and visualization," *Cluster Computing*, vol. 22, pp. 921–929, 2019, doi: 10.1007/s10586-017-1110-2.

[15] S. S. Lad and A. C. Adamuthe, "Improved deep learning model for static PE files malware detection and classification," *International Journal of Computer Network and Information Security*, vol. 14, no. 2, pp. 14–26, 2022, doi: 10.5815/ijcnis.2022.02.02.

[16] M. Imran, M. T. Afzal, and M. A. Qadir, "Similarity-based malware classification using hidden markov model," *4th International Conference on Cyber Security, Cyber Warfare, and Digital Forensics, CyberSec 2015*, pp. 129–134, 2016, doi: 10.1109/CyberSec.2015.33.

[17] T. Mane, P. Nimase, P. Parihar, and P. Chandankhede, "Review of malware detection using deep learning," *Soft Computing for Security Applications*, 2022, pp. 255–262, doi: 10.1007/978-981-16-5301-8_19.

[18] M. Yamauchi, Y. Ohsita, M. Murata, K. Ueda, and Y. Kato, "Anomaly detection in smart home operation from user behaviors and home conditions," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 183–192, 2020, doi: 10.1109/TCE.2020.2981636.

[19] S. Sharma and N. Mishra, "Horizoning recent trends in the security of smart cities: Exploratory analysis using latent semantic analysis," *Journal of Intelligent and Fuzzy Systems*, vol. 46, no. 1, pp. 579–596, 2024, doi: 10.3233/JIFS-235210.

[20] U. Tariq and B. Tariq, "Proactive ransomware prevention in pervasive IoMT via hybrid machine learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 2, pp. 970–982, 2024, doi: 10.11591/ijeecs.v34.i2.pp970-982.

[21] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-Rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware detection issues, challenges, and future directions: a survey," *Applied Sciences*, vol. 12, no. 17, 2022, doi: 10.3390/app12178482.

[22] L. Idouglid, S. Tkatek, K. Elfayq, and A. Guezzaz, "Next-gen security in IIoT: integrating intrusion detection systems with machine learning for industry 4.0 resilience," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 3512–3521, 2024, doi: 10.11591/ijece.v14i3.pp3512-3521.

[23] D. Javeed, M. S. Saeed, I. Ahmad, P. Kumar, A. Jolfaei, and M. Tahir, "An intelligent intrusion detection system for smart consumer electronics network," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 906–913, 2023, doi: 10.1109/TCE.2023.3277856.

[24] R. N. Amarnath and G. Gurulakshmanan, "Cloud-based machine learning algorithms for anomalies detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 1, pp. 156–164, 2024, doi: 10.11591/ijeecs.v35.i1.pp156-164.

[25] P. A. Khan, S. Sharma, I. A. Khan, and P. Singh, "Image detection based technique for shutting down the applications opened in windows operating environment," *Contemporary Issues in Social Sciences*, p. 290.

[26] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3779–3795, 2023, doi: 10.1109/TNNLS.2021.3121870.

[27] Y. Chai, L. Du, J. Qiu, L. Yin, and Z. Tian, "Dynamic prototype network based on sample adaptation for few-shot malware detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4754–4766, 2023, doi: 10.1109/TKDE.2022.3142820.

[28] J. Ding and Y. Wang, "A wifi-based smart home fall detection system using recurrent neural network," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 4, pp. 308–317, 2020, doi: 10.1109/TCE.2020.3021398.

[29] J. Singh and J. Singh, "Assessment of supervised machine learning algorithms using dynamic API calls for malware detection," *International Journal of Computers and Applications*, vol. 44, no. 3, pp. 270–277, 2022, doi: 10.1080/1206212X.2020.1732641.

[30] H. Hindy *et al.*, "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020, doi: 10.1109/ACCESS.2020.3000179.

[31] S. Byun, S. W. Jang, and J. Byun, "Performance evaluation of adaptive offloading model using hybrid machine learning and statistic prediction," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 1, pp. 463–471, 2024, doi: 10.11591/ijeecs.v34.i1.pp463-471.

[32] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, “A survey of deep learning methods for cyber security,” *Information*, vol. 10, no. 4, 2019, doi: 10.3390/info10040122.

[33] A. Panconesi, Marian, and W. Cukierski, “Microsoft malware classification challenge (BIG 2015),” *Kaggle*, 2015. [Online]. Available: https://www.kaggle.com/c/malware-classification

BIOGRAPHIES OF AUTHORS

Vikas Verma is working as Coordinator of School, School of Computer Science and Engineering in Lovely Professional University, Punjab (India). He is having more than 17 years of experience in Academics, Research and Academic Administration. In his previous assignment, he worked as HoD (Computer Science) in Jaipur Engineering College and Research Center, Jaipur. His academic qualifications include B.Tech. (CSE), M.Tech. (CSE) and currently, he is pursuing Ph.D. (CSE). He can be contacted at email: vermavikas2407@gmail.com.

Dr. Arun Malik is working as a Professor in Lovely Professional University, India. He holds M.Tech. and Ph.D. degrees. He has over 13 years of teaching experience and has published around 60 research papers in SCI/Scopus indexed journals and conferences. He has supervised 8 M.Tech., and supervising 8 Ph.D. dissertations. His research interests are wireless networks, adhoc networks (VANET and MANET), internet of things (IoT), and network security. He can be contacted at email: arunmalikhisar@gmail.com.

Dr. Isha Batra is working as a Professor in Lovely Professional University, India. She holds M.E. and Ph.D. degrees. She has over 13 years of teaching experience and has published around 44 research papers in SCI/Scopus indexed journals and conferences. She has supervised 9 M.Tech., and supervising 8 Ph.D. dissertations. Her research interests include wireless networks, sensor networks, ad hoc networks, internet of things, and network security. She can be contacted at email: isha.batra2487@gmail.com.

A. S. M. Sanwar Hosen received the M.S. and Ph.D. degrees in computer science and engineering from Jeonbuk National University, Jeonju, South Korea. He is currently an Assistant Professor with the Department of Artificial Intelligence and Big Data, Woosong University, Daejeon, South Korea. He has published several articles in journals and international conferences. His current research interests include wireless sensor networks, the internet of things, fog-cloud computing, cyber security, artificial intelligence, and blockchain technology. He has been an expert reviewer for IEEE Transactions, Elsevier, Springer, and MDPI journals and magazines. He has also been invited to serve as the Technical Program Committee Member for several reputed international conferences, such as IEEE ACM. He can be contacted at email: sanwar@wsu.ac.kr.