Semiconductor Optoelectronics
ISSN: 1001-5868

AN ENSEMBLE APPROACH LEVERAGING CONVENTIONAL MACHINE LEARNING ALGORITHMS FOR OBFUSCATED MALWARE DETECTION

Lingaraj Sethi author[1*], Dr Prof Prashanta Kumar Patra[2]

[1*]Research Scholar, Computer Science and Engineering, Biju Patnaik University of Technology, Rourkela

[2]Dean,SRIC, Computer Science and Engineering,SOA University Bhubaneswar, Odisha

Abstract

Static and dynamic analysis are the two categories into which malware detection techniques can be divided. Each class's conventional methods have benefits and drawbacks of their own. For instance, although dynamic analysis is slower and needs more resources, it can detect malware variants created through code obfuscation more successfully than static analysis, which is faster but unable to do so. In this research, a novel ensemble model for malware detection is proposed that mitigate above discussed problem. Gradient Boosting (GB), Support Vector Machine (SVM), AdaBoost and Logistic regression (LR) are integrated to form an ensemble model. Initially a dataset known as CIC-Malmem 2022 is used for training and testing of the ensemble model. Term frequency-inverse document frequency (TF-IDF) technique is used to extract vectorized features in malware detection followed by preprocessing of the data. After this the least absolute shrinkage and selection operator(LASSO) tool is used to select the important features from the extracted features. Based on the selected features the ensemble model is trained and tested for performance evaluation. Finally, the result shows that as compared to individual classification of machine learning (ML) model. the classification performed by ensemble model is much accurate as the overall classification accuracy of the ensemble model is 99.99%. The proposed ensemble model is also contrasted with earlier developed hybrid model on the basis of accuracy and result shows that the suggested model outperformed the earlier developed model.

Keywords: Malware detection, Ensemble model, Machine learning, Gradient Boosting, Logistic regression

1. Introduction

Individuals, corporations, and even whole countries are all susceptible to the dangers posed by malicious software in today's era of rapid digital advancement and growing interconnectedness[1]. Malicious software, sometimes known as malware, is a large category of programs designed to inflict harm. These programs range from viruses and worms to Trojans and ransomware[2]. Malware has the potential to cause a wide variety of kinds of damage, which can range from to data violates, financial losses, damage to reputation, and even physical harm. [3]. As a result, it is of the highest necessity to have procedures that are dependable for identifying and avoiding malware[4].Since the introduction of antivirus software, conventional

signature-based security has been a vital tool in the battle against well-known harmful software. The identification of malwares involves using signatures and patterns [5]. Nevertheless, it becomes progressively harder to deploy this method as more and different version of malware are being developed at a faster pace that can quickly avoid signature-based detection techniques [6].

Recently, there has been a move towards ML approaches for malware detection. This shift is in response to the ever-changing threats that are going to lie within the realm of identifying malware. The use of ML technology can serve as a proactive and flexible detection tool, including via simpler mechanisms such as LR, DT, SVM,Random Forest (RF) among others [7,8]. As long as these models are trained on tremendous datasets containing instances of both known malicious software and benign software, they would be able to recognize particular code or behavior characteristics that differentiate between legitimate software and malware [9]. The move towards adopting traditional ML methods for detecting malware is seen as a major step towards battling cybercrime [10].

The paper considers novel approaches in malware detection that take advantage of various features provided by classical machine-learning models. It investigates the challenges involved in creating, testing, and deploying them in real-world circumstances aimed at defending systems from malware attacks. This is important since malware attacks are becoming increasingly sophisticated. These methods significantly increase the accuracy and efficiency of malware detection, even when confronted with threats that are constantly developing over time.

1.1 The Growing Malware Environment

The world of malicious software is always changing, and this is largely due to the reasons why hackers and state-sponsored actors do what they do[11]. Malware is no longer restricted to opportunistic assaults; rather, it has evolved into a sophisticated instrument that is used by people who have the intention of harming[12]. The two primary approaches for classifying malware are static malware, and dynamic malware. Primarily, malware is divided into these two categories. Static malware refers to the initial generation of malicious software, and dynamic malware, is the second generation of malware[13]. The different types of malwares are shown in Figure 1:

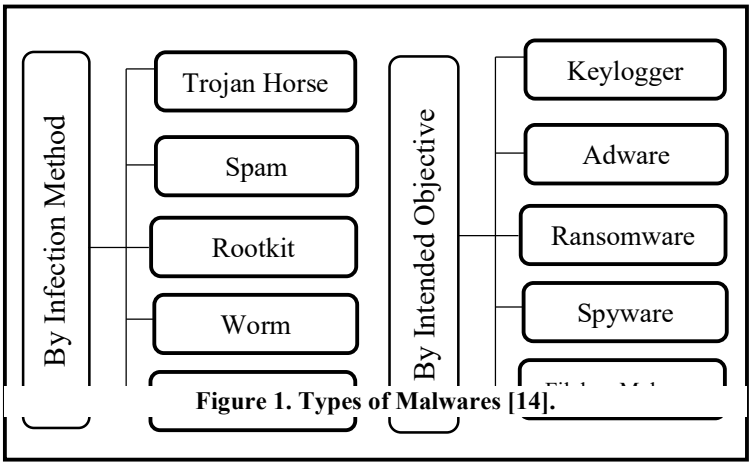


Figure 1. Types of Malwares [14].

From these malware types, some important malwares are defined here:

Trojans: These fraudulent pieces of software pretend to be lawful apps while secretly hiding their ability to do destructive actions. Trojans can penetrate networks and steal sensitive data, give attackers illegal access, or facilitate the installation of other malicious software[15].

Ransomware: Attacks using ransomware have received a substantial amount of attention in recent years. They encrypt the data belonging to a victim and then demand a ransom to unlock it, which can cripple people, businesses, and even vital infrastructure[16].

Worms: Worms propagate swiftly throughout networks, often independent of the actions of individual users. They are a particularly deadly kind of malware because, in addition to being able to reproduce themselves, they can also attack weaknesses in the target system[17].

Spyware: Spyware refers to undetected software that infiltrates a computer with the intention of surveilling user activities and transmitting confidential data to malicious actors. It is possible to utilize it for spying, stealing identities, or other illegal activities[18].

Adware: Adware is a sort of malware that, although not as harmful as other types of malware, nonetheless disturbs the user experience by flooding computers with advertising that the user does not want to see. It can gather user data in certain instances to facilitate targeted advertising[19].

Fileless Malware: This kind of malicious software runs totally in memory and does not leave any traces on the disk. It can avoid detection by conventional means and continue to exist inside a system[20].

Rootkit: Malicious software (malware) to gain access to a computer system, usually at the root level, and maintain lasting control over that system while being undetected by users and security tools is known as a rootkit[21].

Developing effective countermeasures using conventional signature-based approaches has become more difficult as the number of malware variants has become more diverse, and their level of sophistication has increased. As a direct consequence of this, academics and professionals working in the field of security are increasingly looking to ML models to improve their defenses.

1.2 The Promise of ML in Malware Detection

The use of ML, a subfield of artificial intelligence, has shown enormous promise in combating the ever-changing nature of malware. Traditional ML models are intended to do analysis on huge datasets, locate patterns, and formulate predictions based on those findings[22]. When used for the detection of malware, they can automatically adapt to new and previously undiscovered threats by learning from prior data[23].

Conventional ML methods provide a multitude of benefits when used for the detection of malicious software[24]:

Anomaly Detection: These models do very well when it comes to identifying abnormalities in data, which is critical for identifying new and undiscovered strains of malware that do not have recognized signatures[25].

Real-time Detection: Several different ML models are capable of providing real-time detection, which enables an instant reaction to any possible dangers.

Reduced False Positives: ML models can greatly minimize the number of false positives, which are often a problem for signature-based detection systems, by concentrating on behavioral patterns and anomalies.

Adaptability: The ability of ML models to adapt to new strategies and procedures employed by hackers allows these models to become more effective over time as malware continues to undergo development.

2. Literature Review

In this section the previous work of various others in malware detection using different approaches is discussed in detail.

2.1 Machine learning approaches for malware detection.

In (2023)**Roy et al.** [26] presented MalHyStack, a unique hybrid classification approachto identify such network-based obfuscated malware. The suggested working model is built using a deep learning (Extreme Gradient Boosting (XGBoost) Classifier) layer, a layer utilizing traditional ML methods (such as very randomized trees classifier), and a layer of random forest (RF) using a stacked ensemble learning scheme. The suggested method beats the earlier methods on this dataset, according to the experimental findings as a whole.

In (2023)**Alomari et al.** [27]developed a state-of-the-art malware detection system utilizing a combination of deep learning and feature selection methods. Dense and Long-short term memory (LSTM)-based deep learning models are trained using these feature-selected dataset versions. It was shown that, in certain cases, feature selection yielded almost identical results to the original dataset. Rates of decline in the dataset range from 81.77 percent to 93.5 percent, with performance dropping by 3.79 percent to 9.44 percent.

In (2023)**Panda et al.**[28]presented a novel classification ensemble model that takes into account the 25 most salient encoded extracted features from the standard MalImg dataset. Three small-footprint neural network models are stacked in a Stacked Ensemble (SE-AGM): autoencoder, Gated Recurrent Unit (GRU), and multilayer perceptron. Unlike the conventional notion of an ensemble method, the output of one intermediate model is used as input for the following model, which refines the features. SE-AGM is shown to be on par with or better than prior methods, with an average accuracy of 99.43% on the benchmark MalImg dataset.

In (2022)**Masum et al.** [29]introduced a feature-selection-based framework for ransomware detection and prevention that makes use of a variety of MLapproaches to categorize the threat level. To test the efficacy of the suggested approach, authors conducted all tests using a single ransomware dataset. According to the findings of a 10-fold cross-validation trial, the RF classifier consistently beat the earlier of the classifiers in terms of accuracy.

In (2022)**Akhtar, Muhammad Shoaib, and Tao Feng** [30]created an innovative deep-learning technique has been created to tackle the rising tide of malicious software and identify botnet assaults. The system utilizes natural language processing (NLP) techniques as a foundation, combines Convolutional Neural Network (CNN)-LSTM neurons to capture local

spatial correlations, and then learns from the resulting long-term dependencies. The present level of classification accuracy is much higher than 0.81 when compared to the previous study. The CNN-LSTM symmetry correlation shows that, compared to other malware detection techniques like SVM and DT, the detection accuracy is maximized by the CNN-LSTM method, which is at 99%. Accuracy for the rest of the classifiers was somewhere from 98% (DT) to 95% (SVM). The CNN-LSTM model has a perfect F1 score and a 99.9% accuracy rate, 99.9% precision rate, and 99.9% recall rate.

2.2 Deep learning approaches for malware detection

In (2022) **Shatnawi et al.** [31]attempted to detect fraudulent apps by using hybrid approach. While elaborating on the efficacy of these classifiers, authors use deep learning to identify malware activity. Findings show that authorization and the action reiteration feature set are effective in detecting malware in Android apps. The empirical data researchers obtained demonstrate that the accuracy of static, dynamic, and hybrid assessments is extremely near. As a consequence of the research, concluded that static analysis is superior to other methods.

In (2021)**Tian et al.**[32]presented MDCHD, a revolutionary malware detection tool for virtualized settings. First, the Intel Processor Trace (IPT) mechanism is used to record the target program's control flow while it executes. Then it uses control flow data to generate coloured pictures out of it. With these images, deep learning approach based on CNNs is applied to detect malwares from the images. The Lamport's ring buffer algorithm is used in this case for efficiency purposes with respect to the offensive detection methodology. During evaluation, it was found out that this approach can achieve satisfactory levels of detection accuracy with minimum computational costs.

Abusitta et al (2021) [33] suggested an innovative way to detect malware in a dynamic environment. In this approach, deep learning used to find invariant features with respect to ambient changes. This design is based on Denoising Autoencoder as the core for building a customizable deep neural network, which is a type of Autoencoder algorithm. In the end, a practical example with real-world data sets shows much higher performance than the baseline detection.

In (2021)**Basnet et al.,** [34]developed an original framework aimed at detecting ransomware specifically for supervisory control and data acquisition (SCADA) controlled electric vehicle charging stations (EVCS). On simulated cases, each of these three deep learning frameworks maintains an average accuracy of about 97%. At the same time, after conducting ten-fold stratified cross validation, they still have an average F1-score and FAR lower than 1.88%.

In (2020) **Liu et al.**[35]utilized adversarial training and data visualization for deep learning-based detectors. Results from testing the proposed method on Ember malware databases demonstrate that it is able to halt zero-day assaults and achieve an accuracy level of up to 97.73%, with an overall average of 96.25% for all malware types examined.

In (2019)**Feng et al.** [36]presented MobiDroid, a system that uses deep learning to identify Android malware and provide a safe, quick-response environment for mobile devices. MobiDroid can offer a reliable detection accuracy and quick detection solutions on cell phones directly, thanks to well-chosen features and effective feature extraction.

3. Problem formulation

The growing threats from malware, such as trojans, ransomware, and spyware, are mitigated by realizing the limits of traditional signature-based antivirus programs. Cybercriminals' ever-evolving strategies surpass the limitations of these conventional countermeasures, leaving open doors to vulnerabilities including unapproved access, data breaches, monetary losses, and interruptions to services. This problem can be solved by an ensemble model that includes ML traditional techniques such as GB, SVM, LR, AdaBoost. This ensemble approach is intended to improve the ability to detect malware by reducing false positive threshold and facilitating dynamic threat detection. According to this investigation, since these ML models have been used in this research, cybersecurity enterprises can come up with cyber security initiatives with protection of computer networks, systems and sensitive data.

4. Research Methodology

CIC-Malmem 2022 [26] dataset is used in this approach. The next step after that is pre-processing of the data. In preprocessing, the dataset involves cleaning the data, normalizing, and addressing any missing values. This should be done so as to ensure quality of the dataset. The use of TF-IDF approach that aids in vectorization-based feature extraction process ensures better representation of malware features. Consequently, LASSO is employed since it simplifies the dataset and selects the best features for this purpose. In order for the model's efficiency to maximize most important features are retained. Afterwards the data is split into train set and test set to check how well the model works. Furthermore, these ML algorithms are used to train the model on classifying malware instances including GBM, SVM AdaBoost and LR. Therefore, a holistic strategy covering major aspects affecting cybersecurity industry such as data quality, feature representation, model efficiency etc., comes within a method comprising previous methodology and models for enhancing accuracy of malware detection.

The following methods were carried out in this methodology:

- **TF-IDF for Vectorisation-based Feature Extraction**

TF-IDF [37] is an effective method for extracting vectorized features for detecting malwares. For ML algorithms to effectively use it thereby; there must be a translation of textual information related to malware samples into numerical form using TF-IDF. It takes into account both within-document and corpus-wide prevalence of terms. Higher scores on TF-IDF take into account the discriminatory power of phrases that are prevalent in a specific document but not common throughout the dataset at large. This can be used in constructing a feature matrix such that each row show a malware sample, and each column show a distinct phrase, the TF-IDF score being weighted according to its respective importance. The resulting TF-IDF vectorization provides a more accurate representation of malware-related features, thus laying down the foundation for ML models to detect abnormalities implying malicious intentions.

TF, which stands for the number of times a particular phrase looks in a particular text. Another possible application for it is as a percentage of the total number of words in a document. Therefore,

$$TF(t,d) = \frac{a}{b} \tag{1}$$

Where a is the total number of occurrences of t in document d and b is the total number of words. Words' individuality in the corpus is evaluated using a metric called the IDF.

$$IDF(t,d) = log\left(\frac{M}{m}\right) (2)$$

In equation 2, the total number of documents in the corpusis M, and the number of occurrences of the term t in those documentsis m.

$$TF - IDF = TF \times IDF (3)$$

- **Feature Selection using LASSO**

Malware detection relies heavily on a technique called LASSO [38] for selecting features. To improve model efficiency and avoid overfitting, LASSO is used to extract and keep just the most important features from the dataset. This method is essential in malware detection because it promotes sparsity by reducing certain feature coefficients to zero, hence simplifying the feature space. In this way, LASSO picks a subset of characteristics that substantially contribute to differentiating malicious from non-malicious occurrences, resulting in a model that is both easier to read and computationally more efficient. Gradient boosting, support vector machines, AdaBoost, and LR are just a few of the common machine-learning models that can be trained using the features that were previously specified.The overall efficiency of the malware detection system is enhanced because of the incorporation of LASSO into the process of selecting features; this also contributes to the system's increased accuracy and resilience.

The goal function of linear regression is modified by the addition of the LASSO regularization component to promote sparsity. The following is an example of how the objective function for LASSO can be written:

$$min\frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{P}|\beta_j| \quad (4)$$

- λ is the regularization parameter.
- In the dataset, there are n observations.
- There are P features in total.
- For the *i-th* observation, the target variable is *yi*.
- *xij* represents the *i-th* observation's *j-th* feature value.
- β0 is the intercept.
- βj is the coefficient for the *j*-th feature.

The expression $\lambda\sum_{j=1}^{P}|\beta_j|$represents the LASSO regularization term. The goal is to minimize the sum of squared errors ($min\frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2$) along with the sum of the absolute values of the coefficients. Here, λ determines the strength of the regularization. As λ increases, a greater number of coefficients are likely to be precisely zero, effectively performing feature selection. The outlines of the error function and the constraint function are shown in Figure 2. The LASSO is represented by the light blue diamond, while the ridge regression is represented by the light blue disk. These are the two constraint areas.Whilethe

bounds of the least squares error function are shown by the red ellipses [39].

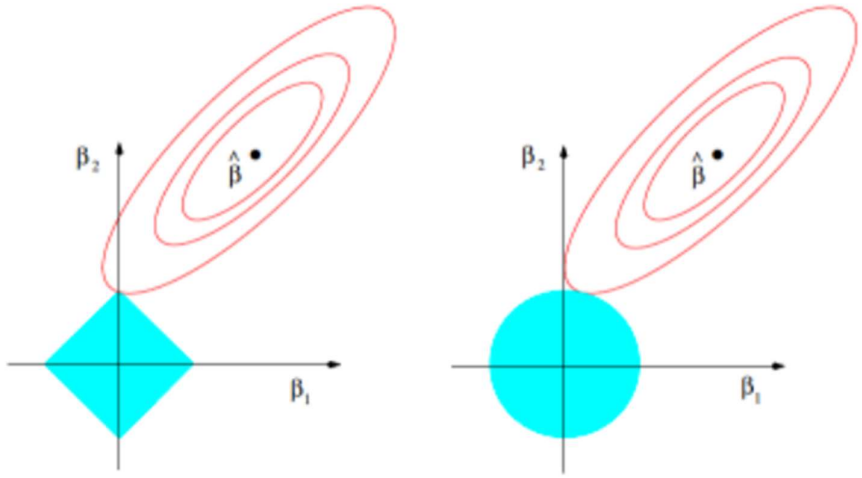


Figure 2.Ridge Regression (right side) and Estimation graph for LASSO (left side) [39].

- **Hybrid Model (GBM+SVM+LR+AdaBoost)**

A hybrid model is used in this instance for the detection of malware. This model combines the advantages of GBM, SVM, LR, and AdaBoost to improve the model's overall performance. During the training phase, you would use the one-of-a-kind capabilities of each algorithm that is included inside the hybrid framework. This would enable a method that is complementary to the categorization of malware. The model can capture a wide variety of patterns and correlations that are present in the data thanks to the ensemble of GBM, SVM, LR, and AdaBoost. This results in a malware detection system that is more reliable and accurate. This hybrid approach intends to leverage the complementary character of individual algorithms, with the end goal of boosting the total efficacy of the malware detection process by using this complementary nature. The incorporation of such a wide variety of models adds to the development of a holistic and cutting-edge strategy for combating the ever-changing threats to cybersecurity.

4.1 Proposed Methodology

In this section, the flowchart of the proposed methodology is given which is shown in Figure 3 below:

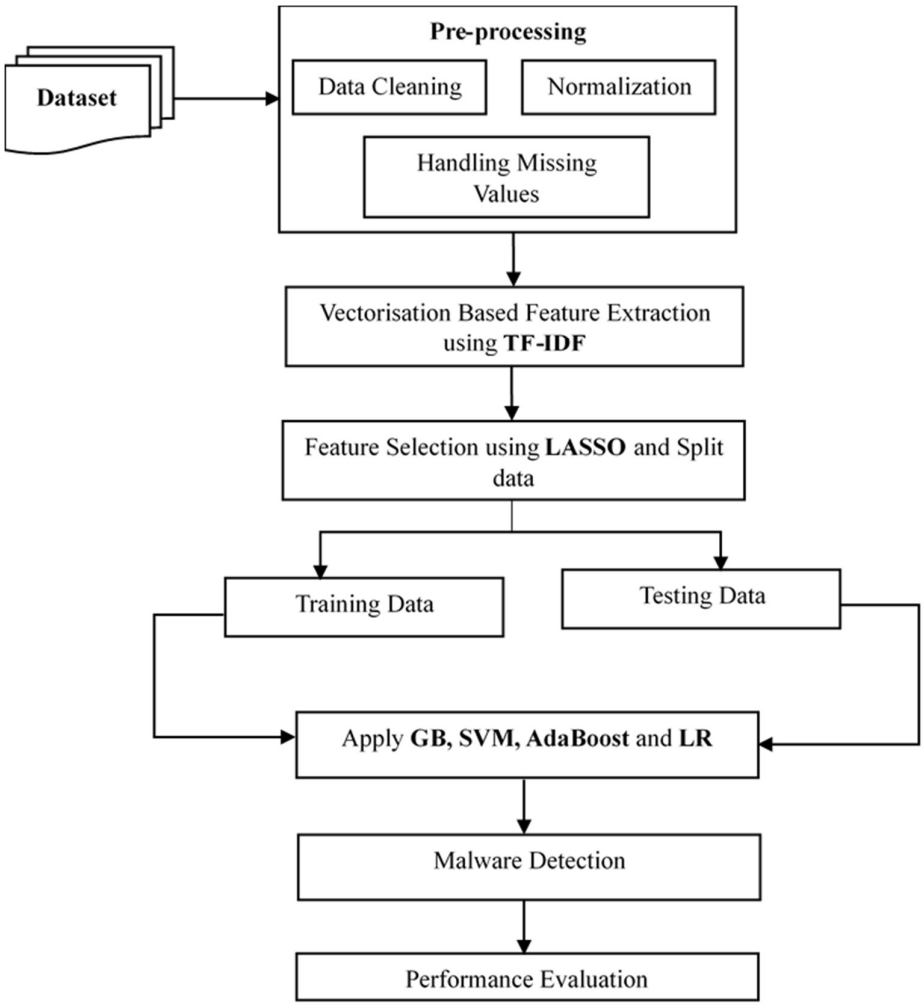


Figure 3. Proposed Methodology.

4.2 Proposed Algorithms

This section provides the proposed algorithm which is given below.

Start

Step 1: Input Dataset:

Let D represent the input Dataset

Step 2:Preprocessing of the data:

1. **Data Cleaning:** Identify and handle outliers, duplicates, and inconsistencies based on domain knowledge.

2. **Normalization:** For each feature X_i in D:

$X_{normlized,i} = \frac{X_i - \text{mean}(X_i)}{std(X_i)}$, where $\text{mean}(X_i)$ is the mean of the feature X_i, and $std(X_i)$ is

---

its standard deviation

3. **Handling Missing Values:** Impute missing values in D using mean, median, or mode.

Step 3:Vectorization-based Feature Extraction using TF-IDF:

For each term t in each document d in D:

$$TF(t,d) = \frac{number\ of\ times\ t\ appers\ in\ D}{Total\ number\ of\ terms\ in\ d}$$

$$IDF(t,d) = log\left(\frac{Total\ number\ of\ document\ in\ D}{number\ of\ document\ containing\ t}\right)$$

$$TF - IDF = TF\ (t,d) \times IDF(t,D)$$

Step 4:Feature Selection using LASSO

Minimize the LASSO objective function:

$$min\frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{P}|\beta_j|$$

Step 5:Splitting Data into Training and Testing Sets:

Split D into training sets as D_{Train} and testing set as D_{Test}.

Step 6:Applying Conventional Machine Learning Models:

1. **Gradient Boosting (GB):**

Train GB on D_{Train}

2. **Support Vector Machines (SVM):**

Train SVM on D_{Train} using the SVM optimization problem.

3. **AdaBoost:**

Train AdaBoost on D_{Train}.

4. **Logistic Regression (LR):**

Train LR on D_{Train}

Step 7:Performance Evaluation:

performance evaluation based on performance evaluation metrics.

End

5. Result and discussion

This section examines the results of implementing the proposed methodology, utilizes these results to establish the performance evaluation criteria for the proposedmodel, and subsequently compares these findings with those of other conventional approaches using the same evaluation criteria to demonstrate the model's resilience.

5.1 Dataset description

The dataset that is used in this research is collected through primary sources and commonly known as CIC Malmem 2022. This dataset is an open-source dataset that is easily available on the website of Kaggle for research purposes.This dataset is formulated by the Canadian institute for cybersecurity, with the vision of malware detection specially malware such as obfuscated. In this dataset there are total 58,596 records in which 29,298 are benign and 29,298 are malicious records [26].

5.2 Evaluation metrics

The efficacy of the proposed model is investigatedutilizing performance parameters such as accuracy, precision, recall, and F1-measure. It is necessary to calculate these measures in order to understand the robustness of the model and it also help to compare the model with existing models for same research. Formulas to calculate the performance parameters that are discussed above are given below:

Accuracy:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ Predictions} \quad (5)$$

Precision:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Recall (Sensitivity):

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

F1-Measure:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

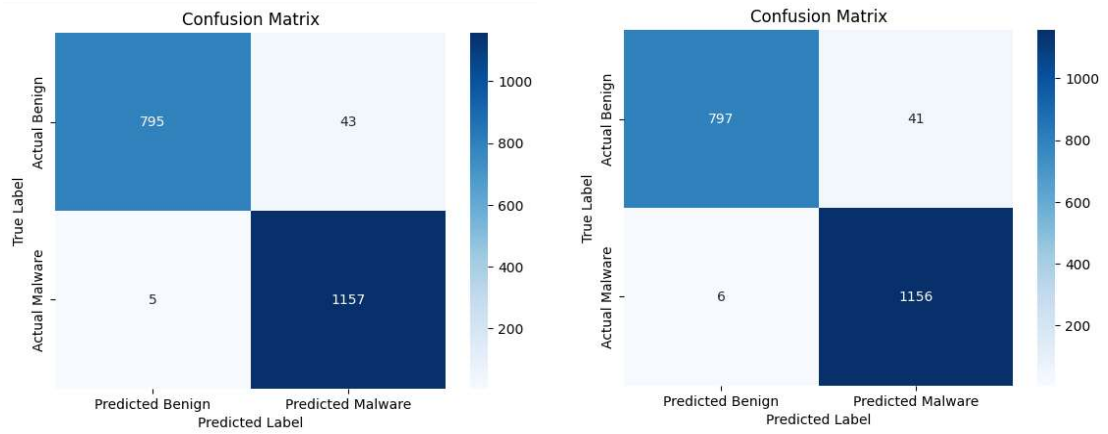
Where TP,FP,FN are true positive, false positive and false negative respectively. Table 1 shows the list of selected features out of total no. of features of the dataset. Based on these selected features the model trained using train set and then the model is tested using test set.

Table 1 List of selected features

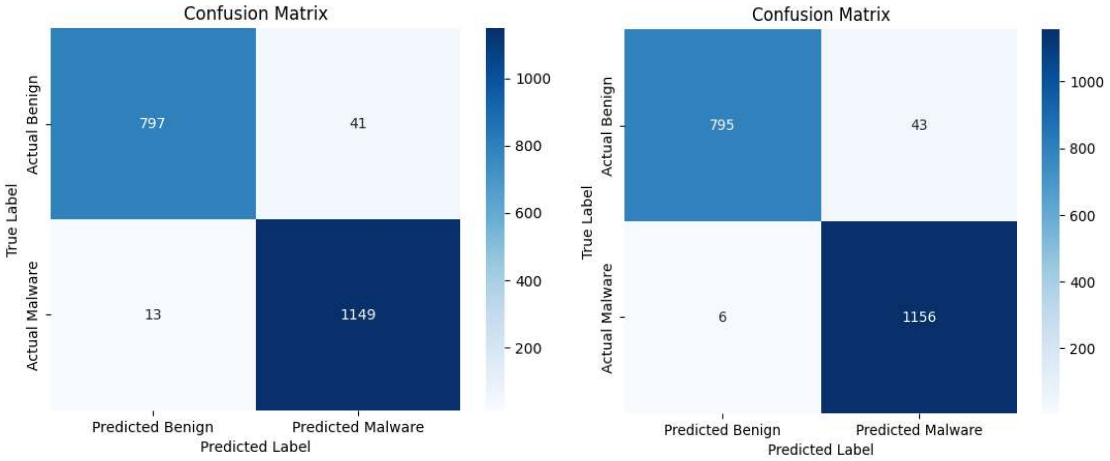
S. No	Selected features	S. No	Selected features	S. No	Selected features
1	pslist.prroc	11	Handles.nsemaphore	21	Svcscan.nservices
2	Pslist.nppid	12	Handles.ntimer	22	Svsccan.nactive

3	Pslist.avg_threads	13	Handles.nsection	23	Callbacks.ncallbacks
4	Dlllist.ndls	14	Handles.nmutant	24	Callbacks.nanonymous
5	Handles.nfile	15	Ldrmodules.not_in_load	25	Callbacks.ngeneric
6	Handles.nevent	16	Ldrmodules.not_in_mem	26	Psxview.not_in_desktrd
7	Handles.ndesktop	17	Malfind.commitCharge	27	Psxview.not_in_session
8	Handles.nkey	18	Malfind.protection	28	Malfind.ninjections
9	Handles.nthread	19	Psxview.not_in_pslist	29	Handles.nhandles
10	Handles.ndirectory	20	Modules.nmodules	30	Pslist.avg_handlers

The efficacy of an approach in any identification strategy is dependent on its score of assessment criteria, known as the confusion matrix. A confusion matrix is a method of expressing the performance of a classification system by giving the essential relative information. Figures 4 and 5 show the confusion matrix for various classifiers and the ensemble model, which has four alternative outputs. On the basis of these four outcomes, the entire results were analyzed employing the four most useful assessment criteria mentioned above.



(a) SVM (b) GB



(c) AdaBoost (d) LR

Figure 4 Confusion matrix of different classifiers

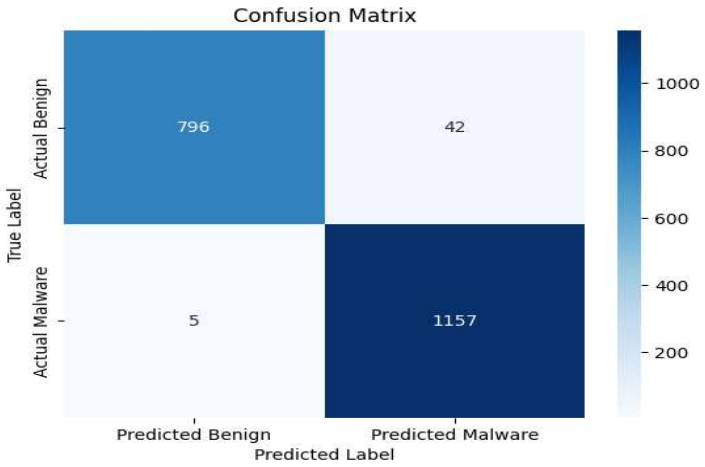


Figure 5 Confusion matrix of ensemble classifiers

Table 2 shows the list of four performance evaluation metrics for all individual ML classifiers and the proposed ensembled model. It is observed that as compared to ML classifiers the ensemble model achieves a higher precision of 99.97%, recall value of 99.88%, f-1 score of 99.95% and accuracy of 99.99%. This shows the efficacy of the proposed ensembled model in detection of malware specially obfuscated malware.

Table 2 List of performance evaluation metrics

Model	**Precision**	**Recall**	**F-1 score**	**Accuracy**
SVM	97.5	97.5	97.5	98.0
LR	97.5	97	97.5	98.0
AdaBoost	97.5	97.0	97.5	97.0

GB	98	97	97.5	98.0
Proposed ensembled model	99.97	99.88	99.95	99.99

5.3 Comparative analysis

Finally, in this section the proposed ensembled model is compared with previously developed techniques based on performance evaluation parameters such as precision, recall, f-1 score and accuracy. And the obtained results are discussed in table 3 also the results are graphically shown in figure 7 as seen below. It is observed that among all the malware detection techniques the proposed ensemble model achieved the highest value in all performance metrics which shows the superiority of the proposed ensemble model in malware detection.

Table 3 Comparison based on performance metrics of various earlier approaches with proposed ensemble model.

Author	**Model**	**Precision**	**Recall**	**F-1 score**	**Accuracy**
Roy et al., [26]	MalHystack	99.97	99.73	99.85	99.85
Alomari et al., [27]	LSTM	94.30	93.70	94.0	94.49
Tian et al., [32]	CNN	95.61	94.97	95.29	95.25
Current study	ensemble model	99.97	99.88	99.95	99.99

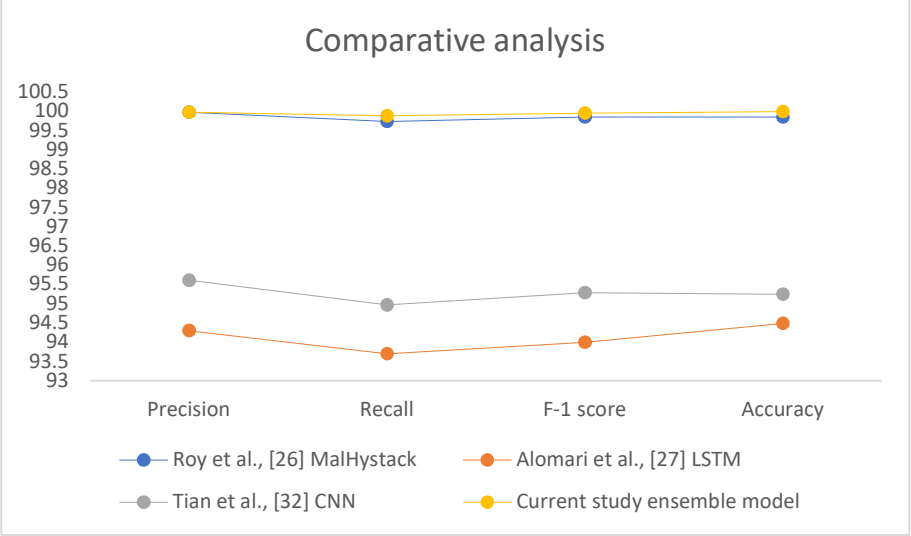


Figure 6Comparison graph

6. Conclusion and future scope

Malware identification is a significant challenge on the Internet of Things (IoT) safeguarding area. The demand for improved malware detection algorithms has increased as the complexity

and diversity of malware has increased. To overcome these challenges this research introduced a novel malware detection model based on ensemble approach. The ML classifier known as GB, LR, SVM and AdaBoost are used to design the ensemble model. The model is capable of performing malware identification tasks with satisfactory results. To enhance the overall accuracy of the proposed ensemble model, the techniques such as TF-IDF and LASSO are used for feature extraction and selection respectively. The finding revealed that the proposed ensemble model achieves an accuracy of 99.99% which is higher than other conventional approaches that are used for classifying binary malware detection. The model's limitation is that it takes longer to execute if there are too many hyperparameters. In future research, to optimize the hyperparameters settings the proposed model is integrated with Bayesian optimization or genetic algorithm which can help to find optimal hyperparameter settings in shorter time period that enhance overall speed of the proposed model.

References

[1]. Frumento, Enrico. "Cybersecurity and the evolutions of healthcare: challenges and threats behind its evolution." M_Health current and future applications (2019): 35-69.

[2]. Maniriho, Pascal, Abdun Naser Mahmood, and Mohammad Jabed Morshed Chowdhury. "A study on malicious software behavior analysis and detection techniques: Taxonomy, current trends and challenges." Future Generation Computer Systems 130 (2022): 1-18.

[3]. Perera, Srinath, Xiaohua Jin, Alana Maurushat, and De-Graft Joe Opoku. "Factors affecting reputational damage to organisations due to cyberattacks." In Informatics, vol. 9, no. 1, p. 28. MDPI, 2022.

[4]. Zotti, Moises, Ericmar Avila Dos Santos, Deise Cagliari, Olivier Christiaens, Clauvis Nji Tizi Taning, and Guy Smagghe. "RNA interference technology in crop protection against arthropod pests, pathogens and nematodes." Pest management science 74, no. 6 (2018): 1239-1250.

[5]. Chakkaravarthy, S. Sibi, Dhamodara Sangeetha, and V. Vaidehi. "A survey on malware analysis and mitigation techniques." Computer Science Review 32 (2019): 1-23.

[6]. Pompura, Mike. "Improved Detection of Multi-Faceted Polymorphic Malware." PhD diss., Florida Institute of Technology, 2021.

[7]. Nazir, Ahsan, Jingsha He, Nafei Zhu, Ahsan Wajahat, Xiangjun Ma, Faheem Ullah, Sirajuddin Qureshi, and Muhammad Salman Pathan. "Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets." Journal of King Saud University-Computer and Information Sciences (2023): 101820.

[8]. Injadat, MohammadNoor, Abdallah Moubayed, Ali Bou Nassif, and Abdallah Shami. "Machine learning towards intelligent systems: applications, challenges, and opportunities." Artificial Intelligence Review 54 (2021): 3299-3348.

[9]. Yan, Jinpei, Yong Qi, and Qifan Rao. "Detecting malware with an ensemble method based on deep neural network." Security and Communication Networks 2018 (2018).

[10]. Hassan, Syed Khurram, and Asif Ibrahim. "The role of Artificial Intelligence in Cyber Security and Incident Response." International Journal for Electronic Crime Investigation 7, no. 2 (2023).

[11]. Broadhead, Stearns. "The contemporary cybercrime ecosystem: A multi-disciplinary overview of the state of affairs and developments." Computer Law & Security Review 34, no. 6 (2018): 1180-1196.

[12]. Caviglione, Luca, Michał Choraś, Igino Corona, Artur Janicki, Wojciech Mazurczyk, Marek Pawlicki, and Katarzyna Wasielewska. "Tight arms race: Overview of current malware threats and trends in their detection." IEEE Access 9 (2020): 5371-5396.

[13]. Sahay, Sanjay K., Ashu Sharma, and Hemant Rathore. "Evolution of malware and its detection techniques." In Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018, pp. 139-150. Springer Singapore, 2020.

[14]. Alenezi, Mohammed N., Haneen Alabdulrazzaq, Abdullah A. Alshaher, and Mubarak M. Alkharang. "Evolution of malware threats and techniques: A review." International journal of communication networks and information security 12, no. 3 (2020): 326-337.

[15]. HosseiniNejad, Reyhaneh, Hamed HaddadPajouh, Ali Dehghantanha, and Reza M. Parizi. "A cyber kill chain based analysis of remote access trojans." Handbook of big data and iot security (2019): 273-299.

[16]. Mayers, Justin. "The Importance of Ransomware Threat Protection & Recovery." PhD diss., Utica College, 2021.

[17]. Ngo, Fawn T., Anurag Agarwal, Ramakrishna Govindu, and Calen MacDonald. "Malicious software threats." The Palgrave Handbook of International Cybercrime and Cyberdeviance (2020): 793-813.

[18]. Kanwar, Akshay Kumar. "An analysis of Key Logger." (2023).

[19]. Vasani, Vatsal, Amit Kumar Bairwa, Sandeep Joshi, Anton Pljonkin, Manjit Kaur, and Mohammed Amoon. "Comprehensive Analysis of Advanced Techniques and Vital Tools for Detecting Malware Intrusion." Electronics 12, no. 20 (2023): 4299.

[20]. Khalid, Osama, Subhan Ullah, Tahir Ahmad, Saqib Saeed, Dina A. Alabbad, Mudassar Aslam, Attaullah Buriro, and Rizwan Ahmad. "An insight into the machine-learning-based fileless malware detection." Sensors 23, no. 2 (2023): 612.

[21]. Mohammadzad, Maryam, and Jaber Karimpour. "Using rootkits hiding techniques to conceal honeypot functionality." Journal of Network and Computer Applications 214 (2023): 103606.

[22]. Sarker, Iqbal H. "Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects." Annals of Data Science (2022): 1-26.

[23]. Al-amri, Redhwan, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A. Al-Sharafi, and Ammar Ahmed Alkahtani. "A review of machine learning and deep learning techniques for anomaly detection in IoT data." Applied Sciences 11, no. 12 (2021): 5320.

[24]. Bharadiya, Jasmin. "Machine Learning in Cybersecurity: Techniques and Challenges." European Journal of Technology 7, no. 2 (2023): 1-14.

[25]. Upman, Vikas, Nikolaj Goranin, and Antanas Čenys. "Convolutional neural network approach for anomaly-based intrusion detection on IoT-enabled smart space orchestration system." In DAMSS 2022: 13th conference on data analysis methods for software systems, Druskininkai, Lithuania, December 1–3, 2022. Vilniaus universitetas, 2022.

[26]. Roy, Kowshik Sankar, Tanim Ahmed, Pritom Biswas Udas, Md Ebtidaul Karim, and Sourav Majumdar. "MalHyStack: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis." Intelligent Systems with Applications 20 (2023): 200283.

[27]. Alomari, Esraa Saleh, Riyadh RahefNuiaa, Zaid Abdi AlkareemAlyasseri, Husam Jasim Mohammed, Nor Samsiah Sani, Mohd Isrul Esa, and Bashaer Abbuod Musawi. "Malware detection using deep learning and correlation-based feature selection." Symmetry 15, no. 1 (2023): 123.

[28]. Panda, Pratyush, Om Kumar CU, Suguna Marappan, Suresh Ma, and Deeksha Veesani Nandi. "Transfer Learning for Image-Based Malware Detection for IoT." Sensors 23, no. 6 (2023): 3253.

[29]. Masum, Mohammad, Md Jobair Hossain Faruk, Hossain Shahriar, Kai Qian, Dan Lo, and Muhaiminul Islam Adnan. "Ransomware classification and detection with machine learning algorithms." In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0316-0322. IEEE, 2022.

[30]. Akhtar, Muhammad Shoaib, and Tao Feng. "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time." Symmetry 14, no. 11 (2022): 2308.

[31]. Shatnawi, Ahmed S., Aya Jaradat, Tuqa Bani Yaseen, Eyad Taqieddin, Mahmoud Al-Ayyoub, and Dheya Mustafa. "An Android malware detection leveraging machine learning." Wireless Communications and Mobile Computing 2022 (2022).

[32]. Tian, Donghai, Qianjin Ying, Xiaoqi Jia, Rui Ma, Changzhen Hu, and Wenmao Liu. "MDCHD: A novel malware detection method in cloud using hardware trace and deep learning." Computer Networks 198 (2021): 108394.

[33]. Abusitta, Adel, Talal Halabi, and Omar Abdel Wahab. "ROBUST: Deep learning for malware detection under changing environments." In AIofAI'21: 1st Workshop on Adverse Impacts and Collateral Effects of Artificial Intelligence Technologies, pp. 1-13. CEUR Workshop Proceedings, 2021.

[34]. Basnet, Manoj, Subash Poudyal, Mohd Hasan Ali, and Dipankar Dasgupta. "Ransomware detection using deep learning in the SCADA system of electric vehicle charging station." In 2021 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America), pp. 1-5. IEEE, 2021.

[35]. Liu, Xinbo, Yaping Lin, He Li, and Jiliang Zhang. "A novel method for malware detection on ML-based visualization technique." Computers & Security 89 (2020): 101682.

[36]. Feng, Ruitao, Sen Chen, Xiaofei Xie, Lei Ma, Guozhu Meng, Yang Liu, and Shang-Wei Lin. "Mobidroid: A performance-sensitive malware detection system on mobile platform." In 2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS), pp. 61-70. IEEE, 2019.

[37]. Somesha, M., and Alwyn R. Pais. "Classification of Phishing Email Using Word Embedding and Machine Learning Techniques." Journal of Cyber Security and Mobility (2022): 279-320.

[38]. Madanan, Mukesh, and Anita Venugopal. "Designing a Hybrid Model Using HSIC Lasso Feature Selection and AdaBoost Classifier to Classify Image Data in Biomedicine." International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies 12, no. 1 (2021): 1-14.

[39]. Kumarage, Prabha M., B. Yogarajah, and Nagulan Ratnarajah. "Efficient feature selection for prediction of diabetic using LASSO." In 2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer), vol. 250, pp. 1-7. IEEE, 2019.