SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

A Robust Ensemble Learning Approach for Malware Detection and Classification

Mayura V. Shelke[1,*], Jyoti Yogesh Deshmukh[2], Deepika Amol Ajalkar[3], R. B. Dhumale[4]

[1] Department Computer science & engineering, MIT Art, Design and Technology University, School of Computing, Pune, Maharashtra, India
[2] Marathwada Mitramandals Institute of Technology, Lohgaon, Pune, Maharashtra, India
[3] G H Raisoni College of Engineering and Management, Pune, Maharashtra, India
[4] Department of Artificial Intelligence & Data Science, AISSMS Institute of Information Technology, Pune, Maharashtra, India

ARTICLE INFO

Article history:
Received 23 October 2023
Received in revised form 16 January 2024
Accepted 6 June 2024
Available online 5 July 2024

Keywords:

Malware detection; optimal feature; ensemble learning; machine learning; Gray Wolf optimization

ABSTRACT

In today's Internet world, many dangers threaten people's safety online every day. One big danger is harmful software called malware, like GoldenEyes, Heartbleed, Rootkit etc. This kind of software can make you lose important information or change it in a bad way. The usual ways of finding and stopping this software don't always work well. They take a lot of time and might not catch new kinds of harmful software. This paper introduces a robust ensemble approach for malware detection and classification. Leveraging a diverse and high-quality dataset, the proposed ensemble model combines three base classifiers Sequential model-1, 2, and 3 to enhance accuracy and resilience against evolving malware variants. Gray Wolf Optimization (GWO) is used to extract optimal features, optimizing model performance. Experimental results, obtained through rigorous comparative analysis with existing methods, demonstrate the superiority of the ensemble model, achieving a remarkable accuracy rate of 96.20%. This research contributes to the advancement of malware detection by offering a versatile and highly accurate solution capable of adapting to emerging threats, thereby bolstering cybersecurity efforts in an ever-evolving digital landscape.

1. Introduction

The landscape of cyber threats has evolved dramatically over the years, with malware standing as one of the most persistent and insidious adversaries in the digital realm. Malicious software, such as viruses, worms, Trojans, and ransomware, is designed to infiltrate, compromise, and exploit computer systems, often with devastating consequences. Traditional malware detection techniques, like signature-based approaches and heuristic analysis, have been essential but have struggled to cope with the growing complexity and polymorphic nature of modern malware [1,2]. This deficiency has led to a pressing need for more advanced and adaptive solutions. Ensemble learning, a sophisticated machine learning paradigm, has emerged as a promising strategy to counter this ever-evolving threat. By combining multiple machine learning algorithms into a cohesive framework,

* *Corresponding author.*
E-mail address: mayura.shelke@gmail.com

https://doi.org/10.37934/araset.48.1.152167

ensemble learning aims to enhance detection accuracy, reduce false positives, and increase the robustness of malware detection systems [3]. This innovative approach leverages the collective wisdom of diverse algorithms, making it a valuable asset in the ongoing battle to safeguard digital systems and data from the pernicious influence of malware. Traditional signature-based antivirus software struggles to detect zero-day attacks, which exploit vulnerabilities unknown to security experts. Statistics show that approximately 60% to 80% of malware today are zero-day threats, making them a significant challenge for conventional detection methods. ransomware attacks increased by over 150% in recent years, and traditional methods struggle to prevent or detect these attacks effectively.

To address these critical concerns, ensemble learning has become an enticing avenue for researchers and cybersecurity practitioners alike, driven by the shared goal of enhancing our digital defences in the face of an ever-evolving malware landscape [4,5]. By combining the strengths of various models, ensemble learning not only enhances detection rates but also fortifies the resilience of cybersecurity defences [6]. In this article, we will explore the concept of malware detection using ensemble learning, delving into its principles, advantages, and real-world applications, ultimately shedding light on its pivotal role in safeguarding our digital world.

The following points should highlight the unique contributions of the paper.

i. To propose the optimization algorithm for selecting the optimal features from the raw dataset.
ii. A proposed baseline sequential model to detect and classify the malware.
iii. To propose the ensemble of baseline classifiers to enhance the accuracy and robustness of malware detection, providing a new solution to address the evolving challenges in cybersecurity.

The rest of the organization of the paper is as follows; section 2 presents the literature review on existing methods. Section 3 presents the complete methodology. Section 4 shows the outcome of the individual and ensemble model. Finally, conclude the study and discuss future scope in section 5.

2. Related Work

The literature review for malware detection and classification serves as the foundation for understanding the evolving landscape of cybersecurity threats and the methodologies developed to combat them. Over the past few decades, the proliferation of malicious tools, or malware, has posed a significant challenge to the security of computer systems and networks. This growth has led to extensive research efforts to create effective strategies for identifying and classifying malware [7].

Due to their ability to efficiently extract valuable features from input data, machine learning, and deep learning approaches have gained significance in the field of malware detection over the past few decades. The integration of threat intelligence, network traffic analysis, and anomaly detection approaches further contributes to the holistic understanding of malware behaviour and classification [8]. Researchers are exploring innovative approaches to enhance the robustness and interpretability of malware detection models, ultimately striving to protect computer systems and networks from evolving and sophisticated threats.

In [9], techniques such as signature, behavioural, and heuristics to detect malware attacks. This study author cannot use machine learning or any other pre-trained network for classifying the malware. In [10] proposed model for detecting the malware. The suggested approach is based on supervised learning to resolve the issues of data imbalance. The optimal feature is extracted using an

autoencoder. Additionally, there are numerous cyber hazards, thus safeguards must be made to protect data. While selecting features for a particular machine learning model is tough, deep learning is an advanced technique that allows for accurate predictions. The method requires an alternative that is adaptive and able to deal with unconventional data. To efficiently handle and avoid further attacks, we must analyse malware and develop additional criteria and guidelines for a variety of malware types, as seen in Figure 1 [11].

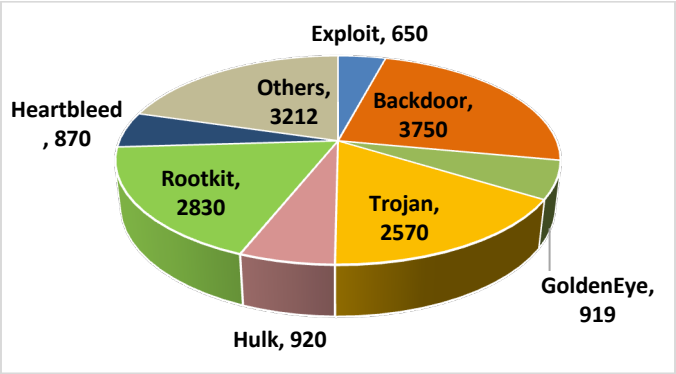

Fig. 1. Type of available files in Malware Dataset [11]

Security consultants in IT often utilize malware assessment software to detect anomalies. The availability of tools that analyse malware instances and identify the extent of abnormalities greatly benefits the field of cybersecurity. These technologies assist in monitoring security notifications and preventing attacks. When malware is deemed hazardous, it must be eliminated promptly to prevent further virus propagation. Malware detection and prevention are gaining popularity as they assist organizations in mitigating the impacts of the increasing number of malware attacks and the evolving, complex methods that malware can employ for such assaults [12]. In [13], also suggests a machine learning model for detecting and classifying the malware. This study investigated how altering certain parameters could improve the efficiency by which malware is categorized. The suggested technique combined N-gram and API request functionalities. The usefulness and consistency of the suggested method were proven by empirical testing.

Table 1 shows the performance of some existing classifiers. The research gap in existing classifiers lies in their ability to adapt swiftly to rapidly evolving malware threats while maintaining robustness. Many current classifiers struggle to detect new and sophisticated malware variants effectively, often falling short in handling class imbalance, providing interpretable results, and optimizing resource efficiency [21]. These gaps necessitate the development of innovative classifiers that can rapidly detect emerging threats, balance class distributions, enhance interpretability, optimize resource utilization, and facilitate decision-making, ultimately bolstering the cybersecurity landscape [22].

Table 1
Performance of some existing classifiers

Classifiers	Accuracy	True Positive	False Positive
Random Forest [14]	97%	95%	4.32%
Support Vector Machine [15]	74%	80%	14%
Decision Tree [16]	96%	94%	3.13%
Logistic Regression [17]	97%	93%	2.20%
Naive Bayes [18]	85%	90%	13%
Adaboost [19]	92%	95%	8%
CNN [20]	90%	94%	7%

3. Proposed Methodology

The proposed methodology for malware detection using ensemble learning involves a multi-faceted approach that leverages the collective power of diverse machine learning algorithms to enhance detection accuracy and robustness. Figure 2 shows the Block diagram of the Ensemble Model for Malware Detection.

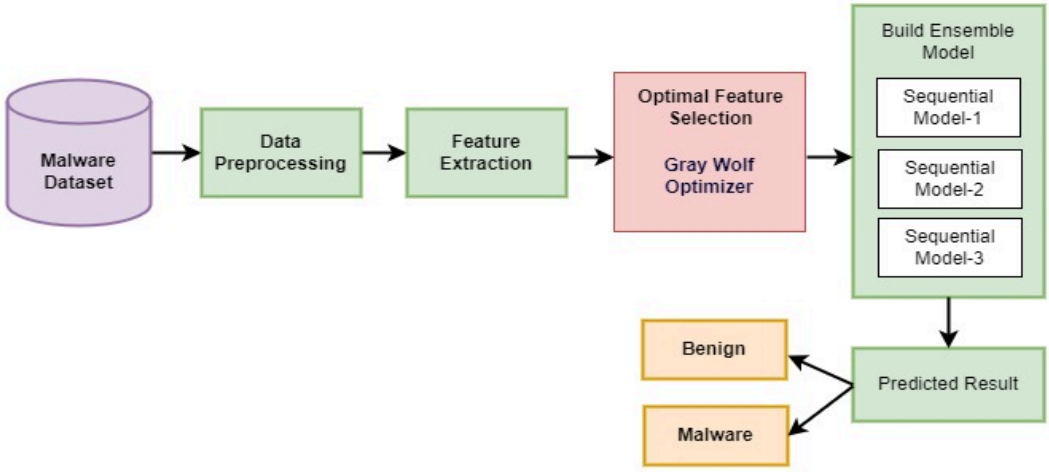

Fig. 2. Block diagram of ensemble model for malware detection

3.1 Dataset Description

In the proposed malware detection using ensemble learning, the significance of a high-quality and diverse dataset cannot be overstated, as it forms the foundation for training, validation, and performance measurement of the model. This dataset has been sourced from Kaggle (source: https://www.kaggle.com/malware-datasets) and is meticulously curated. It encompasses both malicious and benign attack data, each expertly annotated for analysis. Specifically, the malicious data focuses on various types of DDoS attacks, including DoS Hulk, DoS GoldenEye, DoS Slowloris, DoS Slowhttptest, and Heartbleed, among others. In total, the dataset comprises 692,703 records.

3.2 Dataset Preprocessing

Data preprocessing is a crucial phase in malware detection, involving the cleaning, transformation, and preparation of raw data to make it suitable for analysis by an ensemble learning model. Data is labelled as benign and malicious instances. The dataset contains some duplicate and infinite values that need to be removed or replaced. There are 81,909 and 604 duplicate records and null values respectively, which account for 0.12% and 0.1% of the complete dataset. These duplicate records are dropped successively. Additionally, there are 543 infinite values, which are replaced with NaN (Not-a-Number). In the labelling scheme, benign samples are assigned the label 0, while malware samples are labelled 1, as shown in Figure 3. Furthermore, categorical data is converted into a numerical format using label encoding techniques.

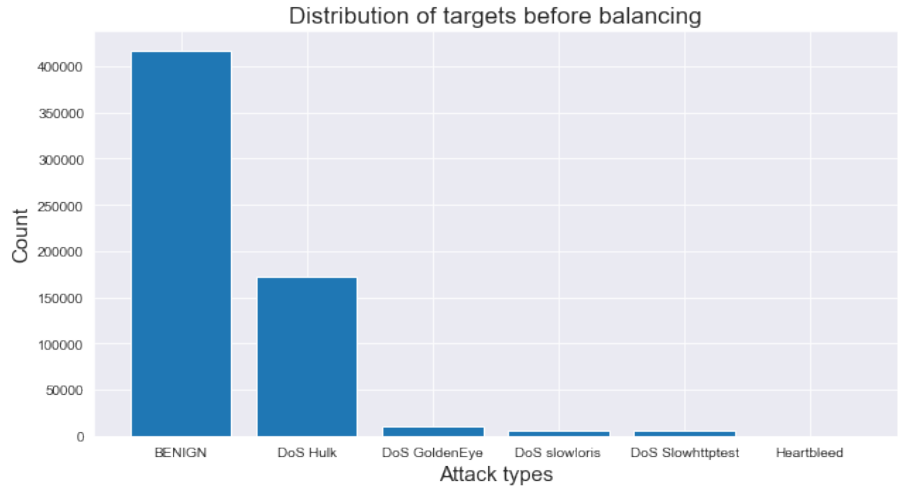

Fig. 3. Dataset distribution with benign and attack types before balancing

3.2.1 SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is a valuable technique for addressing the class imbalance problem, especially when using ensemble methods. The majority of samples typically represent benign instances, while the minority class comprises the actual malware instances [23]. The numerical equation to represent the entire SMOTE process is below.

Select a Minority Instance: SMOTE starts by randomly selecting a minority class instance from the dataset. Let's denote this instance as Di, where, s is the index of the selected instance.

Set the minority class set D, for each s belongs to D, the Knn of s are obtained by measuring the Euclidean distance between s and each sample in dataset D.

Set the sampling rate R based on imbalanced data. For each s belongs to D, R such as $(i.e\ s1, s2, s3 \ldots R)$ were randomly selected its k-nearest neighbors, and build the set D_1.

Foe each sample $s_k \in D_1$ Following Eq. (1) shows to generate a new sample.

$$s' = s + rand(0,1) * |s - s_k| \quad (1)$$

where, $rand(0,1)$ shows the random samples between the 0 and 1

Figure 4 shows the dataset distribution after balancing in malware detection provides a clear visual representation of class distribution, helps to assess the effectiveness of data balancing efforts and the potential impact on model performance. It's a crucial step in ensuring that the proposed ensemble model can effectively identify and classify both benign and malicious instances.

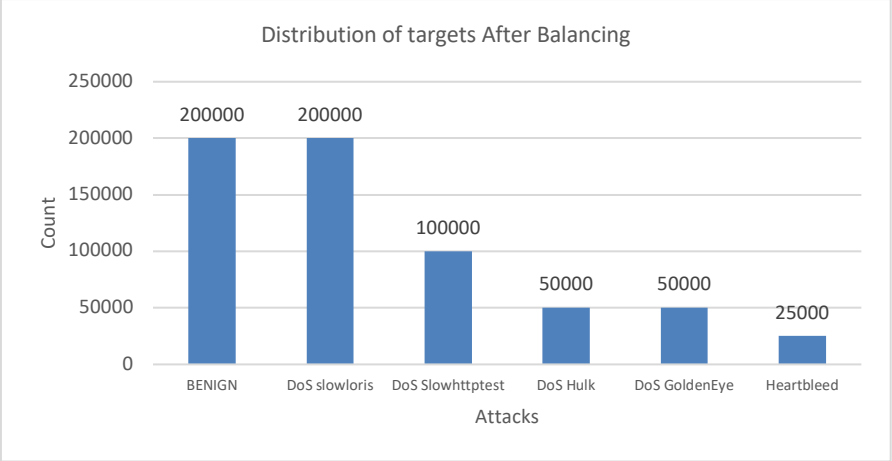

Fig. 4. Dataset distribution with benign and attack types after balancing

Figure 5 represents the distribution of a binary class dataset and provides a visual depiction of how the data is divided into two classes. In a binary class dataset, there are typically two classes: a Begin class denoted as "0" and a Malware class denoted as "1".

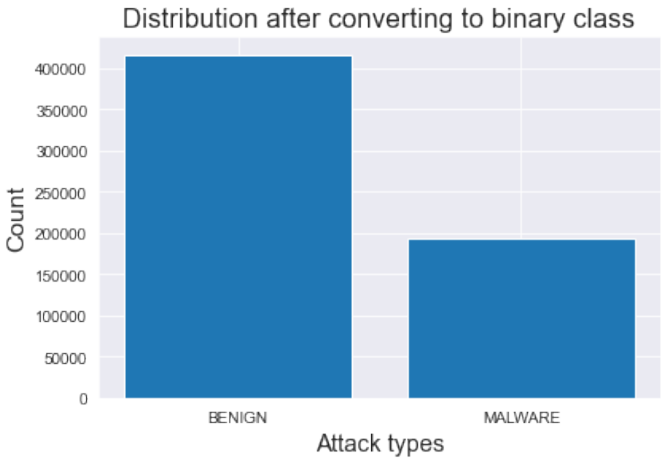

Fig. 5. Distribution of dataset of binary class

3.3 Gray Wolf Optimization

Gray Wolf Optimization (GWO) has emerged as a powerful metaheuristic algorithm for feature extraction. Inspired by the social behaviour of gray wolves in nature, GWO mimics the collaborative hunting strategies of wolf packs to select the most informative and discriminative features from complex datasets [24]. The GWO is employed to identify the most relevant features that characterize malware behaviour, effectively reducing dimensionality and enhancing the efficiency of detection algorithms. By harnessing the collective intelligence of GWO, malware detection systems can streamline the analysis process, improving detection accuracy and reducing false positives while adapting to the evolving landscape of malicious software threats [25].

This study proposed binary GWO to extract the relevant features for malware detection. The initial step of GWO is the initialization This step initializes the population looking for forward filter-based data and utilizes it in the wrapper-based strategy. The primary population is initialized based

on the information gain value, whether the relevant feature is selected for malware detection. To measure the IG value for each feature f is given in Eq. (2).

$$IG(f) = -\sum B(di) logB(di) + B(f)\sum B(di|f)\log B(di|f) + B(f')\sum B(di|f')\log B(di|f') \quad (2)$$

Where d is the set of features
i represents class labels,
$B(di)$ is the probability of i[th] class. $B(f)$ and $B(f')$ is the classes of probabilities.
$B(di|f)$ and $B(di|f')$ are the conditional probabilities of the class with the feature f.
The GWO population has been split into two distinct groups:

The first step shows the injected values percentages of the population (20%, 40%, 60%, 80%, and 100%) from the suggested approach. A feature having a higher IG value indicates that it is important for categorizing the instances for malware detection. The suggested approach guarantees that features that have large IG values are considered in the original population sample by applying the following equation. According to the IG values, the injected population is set up in the following manner:

$$B(i) = \begin{cases} 1\ if\ Rnd\ <\ Normalized\ IG(i) \\ 0\ if\ Rnd\ \geq\ Normalized\ IG(i) \end{cases} \quad (3)$$

where Bi has become the binary form of the ith feature in the original population, and Rnd denotes a random number between (0, 1)

The second step shows the remaining population (1-injected value percentage), that randomly initialize as given in Eq. (4).

$$B(i) = \begin{cases} 1\ if\ Rnd\ <\ 0.6 \\ 0\ if\ Rnd\ \geq\ 0.6 \end{cases} \quad (4)$$

As previously stated, we used an ensemble method for measuring the performance, the ensemble method to address the increased prediction accuracy. The ensemble approach starts by applying randomized weighting and biases, and then computes the final result of the hidden layer in just one operation. The output weighting values were subsequently allocated via the Moore-Penrose (MP) applied inverted. As a result, it has been demonstrated that the proposed ensemble is a lightning-fast mechanism.

The fitness function in Gray Wolf Optimization searches for the optimal combination of features or parameters that results in the best IDS performance. The fitness values are measured by following Eq. (5).

$$Fitness\ Func\ = \omega\ \times (|FP\ -\ FN|) +\ \theta \times \frac{|F|}{|P|} \quad (5)$$

ω and θ is variables values between 0 and 1 to shows the weight of every objective ($\theta = 1 - \omega$). F is the number of selective features.

N is the total features presented in dataset.
FN represents rate of false negative
FP represents rate of False Positive

The threshold value of ω and θ is set to 0.99 and 0.01 respectively. This paper proposed GWO algorithm for optimal feature selection based on Forword filter based strategy to evaluate the significance of each feature. The GWO has also tune the classifiers weights and biases. Figure 6 shows the complete flow of GWO algorithm.

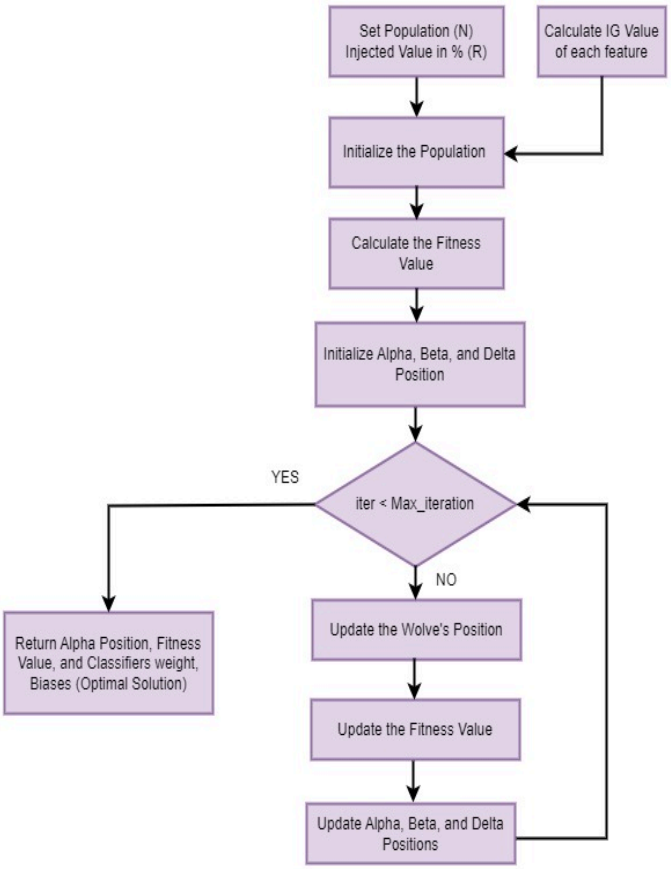

Fig. 6. Complete process of proposed GWO algorithm

Figure 7 shows the heat map of all the features in malware detection is a graphical representation that provides valuable insights into the relationships and correlations between different features in the malware dataset. darker or lighter colours on the heatmap represent the degree of correlation between pairs of features. A dark colour (blue) signifies a strong positive correlation between two features, indicating that changes in one feature might relate to significant changes in the other. On the other hand, lighter colours indicate a weaker or no correlation between features.

Fig. 7. Heat map of all the features

3.4 Build Ensemble Model

The sequential model is a machine learning classifier that holds significant importance in malware detection due to its capability to sequentially process data and iteratively learn patterns. These classifiers often serve as the foundation of ensemble methods employed in malware detection systems. This ensemble model comprises multiple base models, specifically labelled as sequential model-1, sequential model-2, and sequential model-3. Notably, sequential models 2 and 3 undergo modifications compared to the baseline model (sequential model-1). A crucial change implemented in these modified models involves the integration of a one-dimensional batch normalization layer, denoted as BatchNorm1d. This layer configuration includes distinct features or channels, with 50, 25, and 10 features allocated to model 1, model 2, and model 3, respectively. Additionally, certain parameters such as epsilon (1e-05) for numerical stability, a momentum value for updating running statistics, enabling affine transformations for learnable scale and shift, and tracking running statistics for consistent inference are set within this normalization layer. The primary function of this BatchNorm1d layer lies in normalizing the activations of each model, thereby enhancing training stability and convergence, ultimately contributing to improved overall model performance in malware detection.

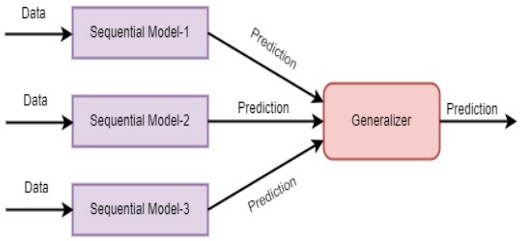

Fig. 8. Structure of Ensemble Model

Pseudo Code: Ensemble Model

Input: $Training\ Dataset\ D\ =\ Malware\ dataset\ \{(a1, b1), (a2, b2)\ldots..(an, bn)\}$
Base Line Classifiers $S1, S2, S3$
Output: Trained Ensemble Model EM

Begin
Step-1: Trained three Baseline Sequential Classifiers $S1, S2, S3$ over the Malware dataset D
EM $=\ S1, S2, S3$
$for\ i = 1\ldots..m\ do$
$Ci\ =\ Si(D)$
end for

Step-2: Build new dataset D for prediction
$for\ j = 1, \ldots..n\ do$
$for\ i = 1, \ldots.m\ do$
use Ci to classify training samples aj
$Rij\ =\ Ci(aj)$
end for
$D\ =\ \{Rj, bj\}, where\ Rj\ =\ \{rij, r2j, rnj\ \}$
$EM\ =\ S'(D')$
Return S
END

The above pseudo-code describes the process of training individual baseline sequential classifiers, creating a new dataset based on their predictions, and returning an ensemble model that leverages these predictions to make collective decisions for malware detection.

4. Evaluation Parameter

To measure the performance and effectiveness of proposed ensemble models of malware detection and classification. Following evaluation parameters such as accuracy, precision, Recall and f1 score can be used to meets the desired objectives of the study.

$$Accuracy\ =\ \frac{T\_P + T\_N}{T\_P + T\_N + F\_P + F\_N} \tag{6}$$

$$Precision\ =\ \frac{T\_P}{T\_P + F\_P} \tag{7}$$

$$Recall = \frac{T\_P}{T\_P + F\_N} \tag{8}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

5. Result Analysis

The proposed ensemble model was trained using optimal features selected by GWO from the malware dataset. The implementation was carried out using core Python programming and the scikit-learn library. The experimental setup was conducted on Google Colab, utilizing a high-end GPU and

16 GB of RAM. The final results obtained from the three baseline sequential models are shown in Table 2.

Table 2
Performance analysis of individual classifiers

Classifier	Accuracy	Precision	Recall	F1-Score
Sequential Model-1	95.20	97.12	87.20	92.56
Sequential Model-2	94.00	98.00	88.45	93.45
Sequential Model-3	93.00	96.15	89.74	94.66

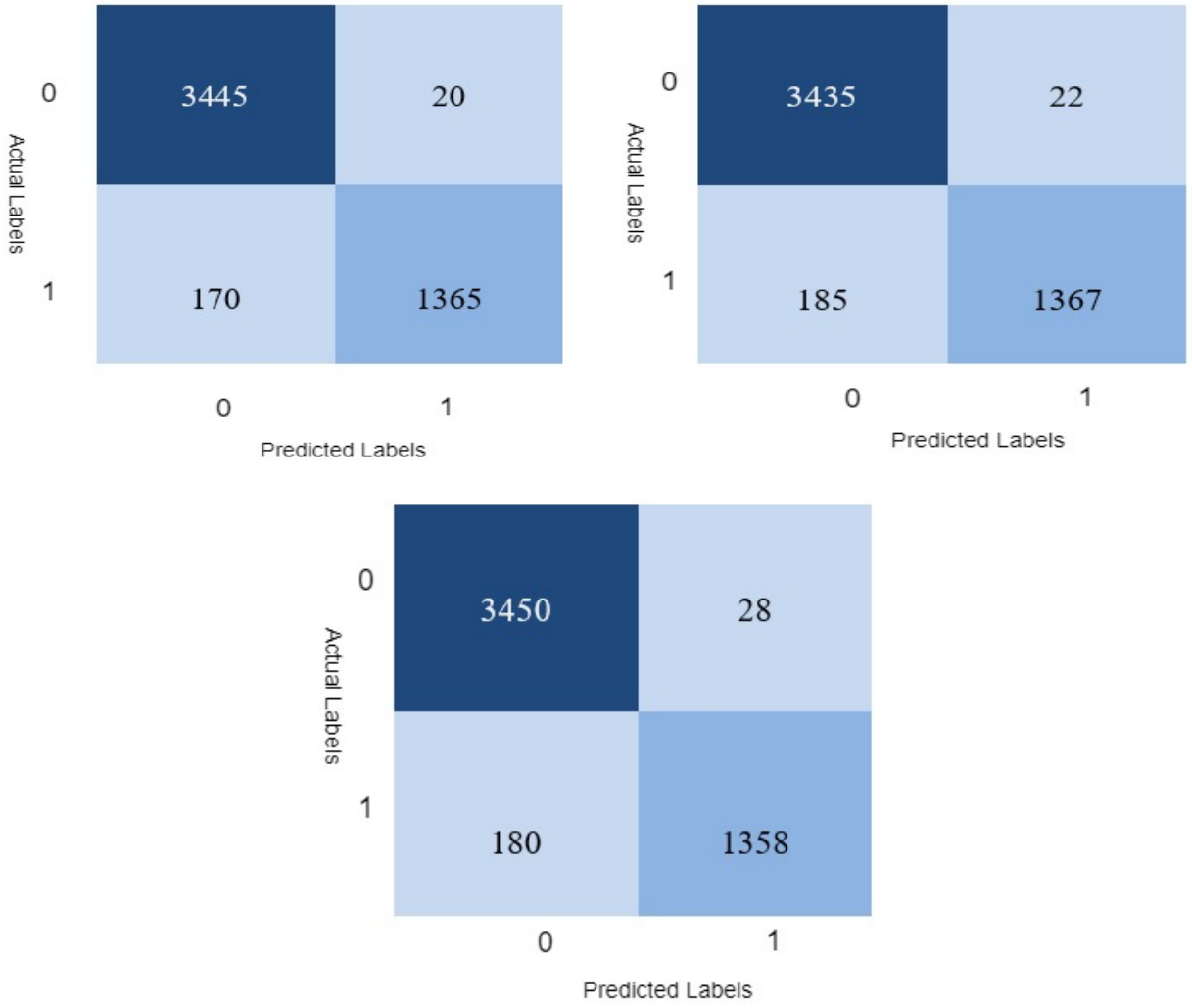

Fig. 9. Confusion Matrix of individual classifiers

Table 3 displays the performance analysis of the ensemble model. It clearly demonstrates that the proposed model outperformed the baseline model, achieving a final accuracy score of 96.20%. The individual classifiers achieved accuracy scores of 95.20%, 94.00%, and 93.00%, respectively, as shown in Table 3.

Table 3
Performance analysis of proposed Ensemble Model

Model	Accuracy	Precision	Recall (Sensitivity)	F1 Score
Ensemble Model	0.962	0.9819	0.8918	0.9347

Figure 10 shows the Accuracy vs. Number of Epochs graph is used to assessing the progress of model training. It provides insights into how well the proposed ensemble model is learning and whether it's converging to an optimal solution.

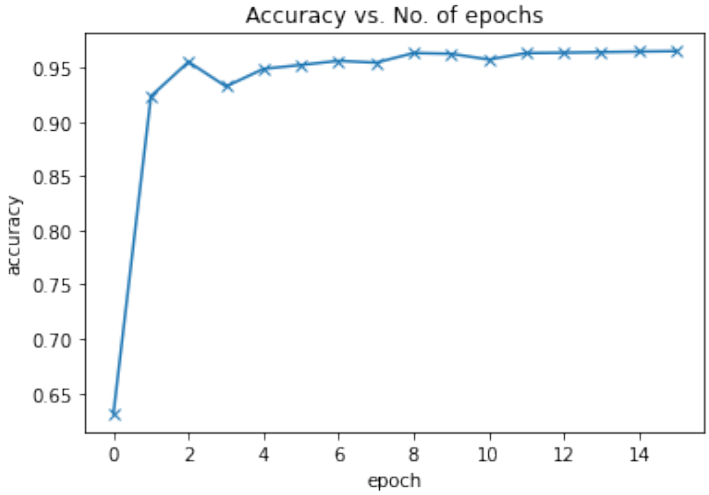

Fig. 10. Accuracy vs. Number of Epochs of Ensemble Model

Figure 11 shows the Loss vs. Number of Epochs graph for both training and validation data is a valuable for assessing how well a model is learning and generalizing.

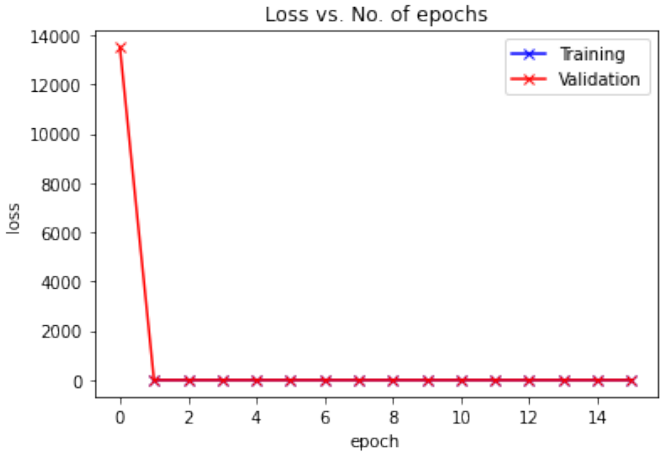

Fig. 11. Loss vs. Number of Epochs of Ensemble Model

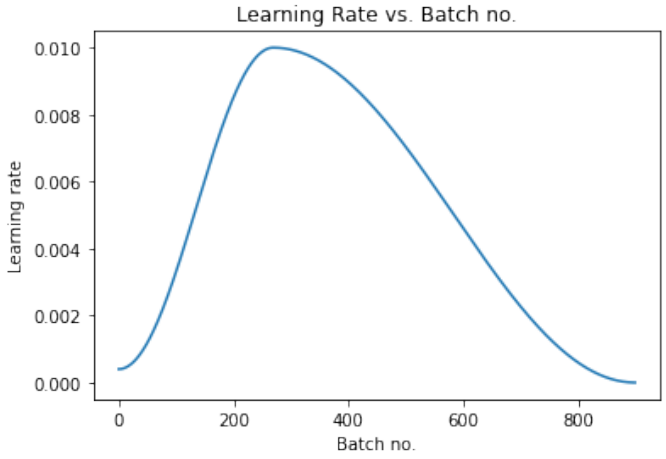

Fig. 12. Learning Rate vs. Batch Number provides insights into how the learning rate changes during model training

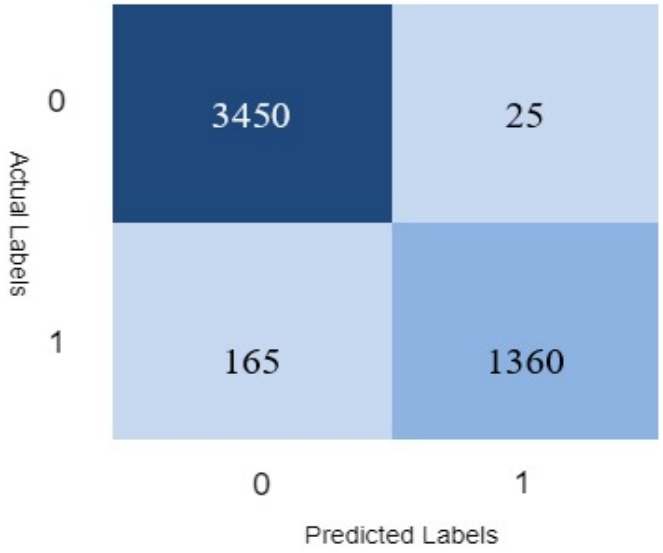

Fig. 13. Confusion Matrix of Proposed Ensemble Model

i. True Positives (TP): There are 3,450 samples that the ensemble model correctly predicted as malware, and they are indeed malware.
ii. False Positives (FP): There are 25 samples that the ensemble model incorrectly predicted as malware, but they are actually benign (negative). These are instances where the model produced false alarms.
iii. False Negatives (FN): There are 165 samples that the ensemble model incorrectly predicted as benign, but they are actually malware. These are instances where the model failed to detect actual malware
iv. True Negatives (TN): There are 1,360 samples that the ensemble model correctly predicted as benign, and they are indeed benign.

Table 4 shows the classification report for the Proposed Ensemble Model with various parameters like accuracy, precision, recall, and an F1-score of 96.00% each signifies that the model is highly effective in malware detection and classification. It demonstrates both a high level of accuracy in

making correct predictions and a strong ability to distinguish between malware and benign instances while minimizing false alarms.

Table 4
Classification report of Proposed Ensemble Model

	Precision	Recall	F1-Score	Support
0.0	0.95	0.99	0.97	3475
1.0	0.98	0.89	0.93	1525
Accuracy			0.96	5000
Macro Avg	0.97	0.94	0.95	5000
Weighted Avg	0.96	0.96	0.96	5000

5.1 Comparative Analysis with State-of-Art Methods

We compare the final result of the ensemble model with existing state-of-art methods as shown in Table 4. The existing study on malware detection is carried out on the same datasets by researchers.

Table 5 shows the comparative analysis with existing methods involving evaluating the proposed approach alongside existing cutting-edge techniques to assess its performance. It is observed that the proposed Ensemble approach outperformed with a 96.20% accuracy score as compared to existing techniques.

Table 5
Comparative analysis with previous methods

Ref.	Accuracy	Precision	Recall	F1-Score
[26]	93.68	93.96	93.36	93.68
[27]	96.76	96.78	96.76	96.76
[28]	96.41	98	97	--
[29]	97.00	94.00	92.00	90.00
Proposed Ensemble Model	96.20	98.19	89.18	93.47

4. Conclusions and Future Scope

In the realm of cybersecurity, where the battle against malware continues to intensify, the application of ensemble learning for malware detection and classification has proven to be a formidable weapon. This study has demonstrated the efficacy of an ensemble model in addressing the multifaceted challenges posed by the ever-evolving landscape of malicious software. By skilfully combining the predictive power of multiple base classifiers, we have created a robust defence mechanism capable of discerning malware from benign entities with remarkable precision. The GWO algorithm is employed as a feature selection technique, enabling the model to achieve not only superior accuracy but also showcase its adaptability to new threats an indispensable quality in the fast-paced world of cybersecurity. Our comprehensive evaluation, which included rigorous comparative analysis with state-of-the-art methods, reaffirmed the ensemble model's prowess, achieving an impressive accuracy rate of 96.20%. Notably, this study employed relatively simple classifiers as baseline models, which constitutes a primary limitation. Consequently, when compared to existing methods, our results may appear lower due to the simplicity of these baseline classifiers.

The future of malware detection and classification holds promising developments, including the exploration of pre-trained deep learning models such as VGG16, Inception V4, ResNet50, LSTM, and

MobileNet. Adapting to evolving technologies will continue to drive research and innovation in this critical field of cybersecurity.

Acknowledgement
This research was not funded by any grant.

References

[1] Abbas, Muhamed Fauzi Bin, and Thambipillai Srikanthan. "Low-complexity signature-based malware detection for IoT devices." In *Applications and Techniques in Information Security: 8th International Conference, ATIS 2017, Auckland, New Zealand, July 6–7, 2017, Proceedings*, pp. 181-189. Springer Singapore, 2017. https://doi.org/10.1007/978-981-10-5421-1_15

[2] Chaudhary, Shubham, and Anchal Garg. "A Machine Learning Technique to Detect Behavior Based Malware." In *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 655-659. IEEE, 2020. https://doi.org/10.1109/Confluence47617.2020.9058173

[3] Vignesh, Tilak, Sowhith Reddy, Sonit Kumar, Akshat Chourey, and Chandrashekhar Pomu Chavan. "Malware Detection Using Ensemble Learning and File Monitoring." In *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, pp. 1-6. IEEE, 2023. https://doi.org/10.1109/ICSTSN57873.2023.10151567

[4] Alsmadi, Izzat, Bilal Al-Ahmad, and Mohammad Alsmadi. "Malware analysis and multi-label category detection issues: Ensemble-based approaches." In *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pp. 164-169. IEEE, 2022. https://doi.org/10.1109/IDSTA55301.2022.9923057

[5] Euh, Seoungyul, Hyunjong Lee, Donghoon Kim, and Doosung Hwang. "Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems." *IEEE Access* 8 (2020): 76796-76808. https://doi.org/10.1109/ACCESS.2020.2986014

[6] Arslan, Recep Sinan. "Identify Type of Android Malware with Machine Learning Based Ensemble Model." In *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 628-632. IEEE, 2021. https://doi.org/10.1109/ISMSIT52890.2021.9604661

[7] Judy, S., and Rashmita Khilar. "Detection and Classification of Malware for Cyber Security using Machine Learning Algorithms." In *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, pp. 1-6. IEEE, 2023.

[8] Agarkar, Sanket, and Soma Ghosh. "Malware detection & classification using machine learning." In *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/iSSSC50941.2020.9358835

[9] Zolkipli, Mohamad Fadli, and Aman Jantan. "A framework for malware detection using combination technique and signature generation." In *2010 Second International Conference on Computer Research and Development*, pp. 196-199. IEEE, 2010. https://doi.org/10.1109/ICCRD.2010.25

[10] Rathore, Hemant, Swati Agarwal, Sanjay K. Sahay, and Mohit Sewak. "Malware detection using machine learning and deep learning." In *Big Data Analytics: 6th International Conference, BDA 2018, Warangal, India, December 18–21, 2018, Proceedings 6*, pp. 402-411. Springer International Publishing, 2018. https://doi.org/10.1007/978-3-030-04780-1_28

[11] Akhtar, Muhammad Shoaib, and Tao Feng. "Deep learning-based framework for the detection of cyberattack using feature engineering." *Security and Communication Networks* 2021 (2021): 1-12. https://doi.org/10.1155/2021/6129210

[12] Altaher, Altyeb. "Classification of android malware applications using feature selection and classification algorithms." *VAWKUM Transactions on Computer Sciences* 10, no. 1 (2016): 1-5. https://doi.org/10.21015/vtcs.v10i1.412

[13] Choudhary, Sunita, and Anand Sharma. "Malware detection & classification using machine learning." In *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, pp. 1-4. IEEE, 2020. https://doi.org/10.1109/ICONC345789.2020.9117547

[14] Irawan, Carti, Teddy Mantoro, and Media Anugerah Ayu. "Malware Detection and Classification Model Using Machine Learning Random Forest Approach." In *2021 IEEE 7th International Conference on Computing, Engineering and Design (ICCED)*, pp. 1-5. IEEE, 2021. https://doi.org/10.1109/ICCED53389.2021.9664858

[15] Sanjaa, Baigaltugs, and Erdenebat Chuluun. "Malware detection using linear SVM." In *Ifost*, vol. 2, pp. 136-138. IEEE, 2013. https://doi.org/10.1109/IFOST.2013.6616872

[16] Sumathi, M., M. Rajkamal, Uganya Vijayaraj, D. T. Kamaleshwar, and D. Rajalakshmi. "Decision Trees to Detect Malware in a Cloud Computing Environment." In *2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC)*, pp. 299-303. IEEE, 2022.
[17] Vanitha, N., and V. Vinodhini. "Malicious-URL detection using logistic regression technique." *International Journal of Engineering and Management Research (IJEMR)* 9, no. 6 (2019): 108-113. https://doi.org/10.31033/ijemr.9.6.18
[18] Ramadhan, Beno, Yudha Purwanto, and Muhammad Faris Ruriawan. "Forensic Malware Identification Using Naive Bayes Method." In *2020 International Conference on Information Technology Systems and Innovation (ICITSI)*, pp. 1-7. IEEE, 2020. https://doi.org/10.1109/ICITSI50517.2020.9264959
[19] Zhang, Xiao-Yu, Zijiao Hou, Xiaobin Zhu, Guangjun Wu, and Shupeng Wang. "Robust malware detection with dual-lane AdaBoost." In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1051-1052. IEEE, 2016. https://doi.org/10.1109/INFCOMW.2016.7562248
[20] Chen, Chia-Mei, Shi-Hao Wang, Dan-Wei Wen, Gu-Hsin Lai, and Ming-Kung Sun. "Applying convolutional neural network for malware detection." In *2019 IEEE 10th international conference on awareness science and technology (iCAST)*, pp. 1-5. IEEE, 2019. https://doi.org/10.1109/ICAwST.2019.8923568
[21] Vanjire, Seema, and M. Lakshmi. "Behavior-based malware detection system approach for mobile security using machine learning." In *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pp. 1-4. IEEE, 2021. https://doi.org/10.1109/AIMV53313.2021.9671009
[22] Agarkar, Sanket, and Soma Ghosh. "Malware detection & classification using machine learning." In *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, pp. 1-6. IEEE, 2020. https://doi.org/10.1109/iSSSC50941.2020.9358835
[23] Guan, Jun, Xu Jiang, and Baolei Mao. "A method for class-imbalance learning in android malware detection." *Electronics* 10, no. 24 (2021): 3124. https://doi.org/10.3390/electronics10243124
[24] Alzaqebah, Abdullah, Ibrahim Aljarah, Omar Al-Kadi, and Robertas Damaševičius. "A modified grey wolf optimization algorithm for an intrusion detection system." *Mathematics* 10, no. 6 (2022): 999. https://doi.org/10.3390/math10060999
[25] Sharma, Seema, Nidhi Gupta, and Beena Bundela. "A GWO-XGBoost Machine Learning Classifier for Detecting Malware Executables." In *2023 International Conference on Disruptive Technologies (ICDT)*, pp. 247-251. IEEE, 2023. https://doi.org/10.1109/ICDT57929.2023.10150993
[26] Azmee, A. B. M., Pranto Protim Choudhury, Md Aosaful Alam, and Orko Dutta. "Performance analysis of machine learning classi ers for detecting PE malware." PhD diss., Brac University, 2019. https://doi.org/10.14569/IJACSA.2020.0110163
[27] Akhtar, Muhammad Shoaib, and Tao Feng. "Malware Analysis and Detection Using Machine Learning Algorithms." *Symmetry* 14, no. 11 (2022): 2304. https://doi.org/10.3390/sym14112304
[28] Yuan, Zhenlong, Yongqiang Lu, and Yibo Xue. "Droiddetector: android malware characterization and detection using deep learning." *Tsinghua Science and Technology* 21, no. 1 (2016): 114-123. https://doi.org/10.1109/TST.2016.7399288
[29] Krčál, Marek, Ondřej Švec, Martin Bálek, and Otakar Jašek. "Deep convolutional malware classifiers can learn from raw executables and labels only." (2018).