International Journal of Computer Science and Engineering (IJCSE)
ISSN (P): 2278–9960; ISSN (E): 2278–9979
Vol. 13, Issue 2, Jul–Dec 2024; 73–86
© IASET

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

A COMPARATIVE PERFORMANCE ANALYSIS OF INTRUSION DETECTION AND MALWARE CLASSIFICATION USING 1D-CNN, TRANSFER LEARNING, AND ENSEMBLE TECHNIQUES

Krishna Kumar & Hardwari Lal Mandoria

G. B. Pant University of Agriculture and Technology, Pantnagar, Uttarakhand, India

ABSTRACT

The critical role of network intrusion detection systems (NIDS) and real-time malware analysis is to safeguard the security and stability of networks and sensitive data across diverse industries, including enterprise, government, IoT, and healthcare sectors. It explores the effectiveness of deep learning approaches, specifically 1D CNN, transfer learning, and ensemble techniques, for malware detection and classification. The experimental work demonstrates that visualization-based methods utilizing convolutional neural networks can efficiently analyze malware images. This research underscores the necessity for updated and novel malware datasets to address the detection of emerging malware types. A 1D CNN and ensemble models were employed for the classification of the well-known real-time gray scale image dataset, Malimg. Additionally, a 2D CNN model based on transfer learning and ensemble techniques is used for the classification of a novel malware RGB image dataset. The performance evaluation of various models revealed that the transfer learning and ensemble technique significantly enhanced accuracy, achieving a peak malware detection rate of 98.83%.

KEYWORDS: *Intrusion Detection System, Ensemble Technique, Transfer Learning, Malware Detection, Malware Classification.*

Article History

Received:* *04 Sep 2024* | *Revised:* *04 Oct 2024* | *Accepted:* *08 Oct 2024

INTRODUCTION

The rapid growth of cyber threats has necessitated advanced intrusion detection systems and malware classification techniques. As technology usage expands, cyber-attacks have increased exponentially worldwide[1]. Traditional machine learning (ML) methods have shown success in detecting unknown malware in real-time, while deep learning approaches can eliminate feature engineering[2]. Intrusion detection systems face challenges in recognizing sophisticated and hidden malware, including zero-day attacks. Recent research has focused on developing innovative ML algorithms to enhance accuracy, efficiency, and adaptability in intrusion detection. These advancements aim to address resent challenges and offer more effective solutions against evolving cyber threats.

Traditional systems face significant challenges in detecting complex malware and intrusions[3]. Sophisticated attacks employ various evasion techniques, including obfuscation, fragmentation, and code reuse, making detection increasingly difficult. Traditional methods like signature-based and heuristic analysis have limitations in detecting new threats, particularly zero-day attacks and file-less malware[4]. The rapid evolution of malware and its diverse behaviours further complicate detection efforts [5]. Intrusion detection systems (IDS) struggle to keep pace with advanced cyber-attacks, which can compromise data confidentiality, integrity, and availability[6][7]. The researchers are exploring new approaches such as real-time detection, sandboxing, and improved feature representation methods. However, the cyber security continuously facing obstacles in effectively countering emerging threats and evasion techniques, necessitating ongoing research and development of more robust detection strategies. Present the need for advanced techniques to improve detection and classification accuracy.

Problem Statement

Artificial intelligence (AI) and machine learning (ML) are increasingly integrated into interpersonal interactions and business environments [8]. However, this technological advancement also raises significant threats and security risks to human life, particularly when technology is misused. Sophisticated malware, viruses, and malicious code are being developed using advanced intelligence, allowing them to evade detection by conventional malware detection systems. Various ML and deep learning (DL) techniques exist for detecting and classifying malware, each with associated computational costs, resource consumption, storage requirements, and training times based on existing input data. However, these techniques often face limitations related to accuracy, processing speed, and the need for substantial computational resources, particularly dealing with large and real-time datasets.

Objectives and Contributions

The primary objective of this paper is to analyze various visualization-based intrusion detection techniques for malware detection and classification. A key contribution of this paper is to compare the performance of malware detection and classification models based on the 1D-CNN, transfer learning, and ensemble technique. It provides an analytical review of existing research and compares it with the results obtained from visualization-based malware detection and classification techniques such as 1D-CNN, ensemble methods, and transfer learning.

Organization of the Paper

The remainder of the paper is arranged as the recent related works are given in Section 2. Section 3 discuss about the proposed methodology. Section 4 illustrates the experimental analysis and results obtained, and the results and discussion are given in Section 5. Finally, Section 6 concludes the paper with suggestions for future work.

Related Works

Recent research has explored deep learning and transfer learning approaches for malware classification using image-based representations of malware. Multiple studies have adapted pre-trained models like VGG16, Xception, and InceptionResNetV2 for this task, achieving high accuracies of 95-98% on various malware datasets. Transfer learning techniques have been employed to leverage features learned from large image datasets, reducing the need for extensive malware-specific training data. Some researchers have investigated data augmentation through code obfuscation to expand limited malware datasets [9]. Additionally, studies have compared the performance of different pre-trained models, including AlexNet, VGG19, and ResNet, for malware image classification [10]. These approaches have shown the ability of DL and transfer learning in improving malware classification accuracy and efficiency.

- Natraj *et al.*[11] developed a classification model that get the accuracy of 97.18% in classifying 25 malware families from9,458 malware sample dataset. Their approach involved using GIST[12] to extract texture features and employing the k-nearest neighbor (k-NN) algorithm with Euclidean distance for classification. Additionally, they utilized 10-fold cross-validation to evaluate the model's performance.

- Vinay kumar *et al.*[13] introduced a deep learning-based, two-stage malware detection framework designed to first detect malware and then classify it into specific types. The framework utilizes a 1D-CNN combined with LSTM (long short-term memory) and was trained on both the highly imbalanced Malimg dataset and a custom private dataset. The model achieved classification accuracies of 96.3% on the Malimg dataset and 98.8% on the private dataset.

- Go *et al.*[14] introduced the ResNeXt50 model for malware classification, achieving an accuracy of 98.32% on the unbalanced Malimg dataset and 98.86% on a custom dataset. The model was trained using 2D grayscale malware images, which required significant computational resources and extended training time.

- Aslan *et al.*[15] proposed an innovative deep learning architecture for classifying various malware variants and families using a hybrid model and a vision-based CNN approach. Their method employs visualization techniques that convert the malware's binary code into fixed-size grayscale images. These images are fed into a hybrid CNN model that integrates ResNet50, AlexNet, and Inception-v3 to extract features. The model achieved classification accuracies of 94.88% on the Microsoft Big 2015 dataset, 96.5% on the Malevis dataset, and 97.78% on the Malimg dataset.

- Awan *et al.*[16] introduced an improved CNN model that integrates spatial attention mechanisms, such as dynamic spatial convolution and VGG19 for feature extraction, while employing base layer freezing. Their experiments, performed on the Malimg dataset containing 25 malware classes, utilized 2D convolution layers 97.68% classification accuracy.

- O'Shaughnessy and Sheridan[17] developed an image-based hybrid framework to address challenges in malware classification. Their approach introduces a novel space-filling curve to extract visual features for multi classification on a dataset of 13,599 samples, including both obfuscated and non-obfuscated malware from 23 families, achieving a classification accuracy of 97.6%. The space-filling curve, a mathematical technique for mapping 1D data into 2D space, converts binary executable files (represented as 1D byte sequences) into 2D grayscale images. The hybrid framework operates in three phases: (1) malware conversion, (2) feature extraction, and (3) classification.

- Lu *et al.*[18] introduced a self-attentive model for real-time malware classification, combining Vision Transformers with CNN to improve accuracy while reducing inference latency. This ensemble approach achieved a top accuracy of 98.17%. Although Vision Transformers demand greater computational resources and extended training time, the proposed model offers a more efficient alternative without sacrificing performance.

- Seneviratne *et al.*[19] utilized a self-supervised DL model that incorporates Vision Transformers (ViTs) [20] for malware detection. Vision Transformers offer a scalable alternative for processing sample images by splitting them into smaller patches (e.g., 8x8 or 16x16 pixels). These patches are embedded with additional information using a transformer encoder, which then decodes them to recreate the original image with reduced feature representations. The proposed model, SHERELOCK, achieved 97% accuracy in malware detection and 87% precision in classifying malware on the MalNet dataset [21].
- Zhong *et al.*[22] proposed a classification framework consisting of three key components: a converter, a feature engineer, and a classifier. The converter transforms the binary files into an image, while the feature engineer applies contrast-limited adaptive histogram equalization to enhance local contrast across different regions of the image. After this, the processed image is resized to a smaller, fixed size to expedite the classification process. The classifier, a shallow CNN-based model is able to classify with accuracy of 96%.
- El-Sayed *et al.*[23] proposed seven image-based malware classification algorithms, evaluating their performance based on accuracy and model complexity. They converted PCAP files into colored images to capture the structure and patterns for analysis. Among the classifiers, VGG16 achieved the optimal accuracy of 96%, followed by SVM 94%. This image-based approach demonstrates its effectiveness in detecting network intrusions caused by both known and unknown malware, while also highlighting potential for further improvements in accuracy.
- Al-Qadasi*et al.*[24] introduced the advanced ConvNeXtV1 and V2 models designed to achieve higher accuracy on large-scale image datasets. While these models excel in classification performance across various image-based datasets, they demand high computational resources to train.

The existing literature indicates that visualization-based approaches for malware classification typically utilize both 1D-CNN and 2D-CNN models. While 2D images necessitate considerable computational resources and training time, 1D-CNNs can achieve comparable classification accuracy with significantly lower resource requirements. This paper evaluates the performance of 1D-CNN models, which have been adapted into a one-dimensional architecture, along with the transfer learning and ensemble models.

PROPOSED METHODOLOGY

This section describes the layout used to analyze the visualization-based malware classification techniques, including the dataset and performance metrics used for performance evaluation.

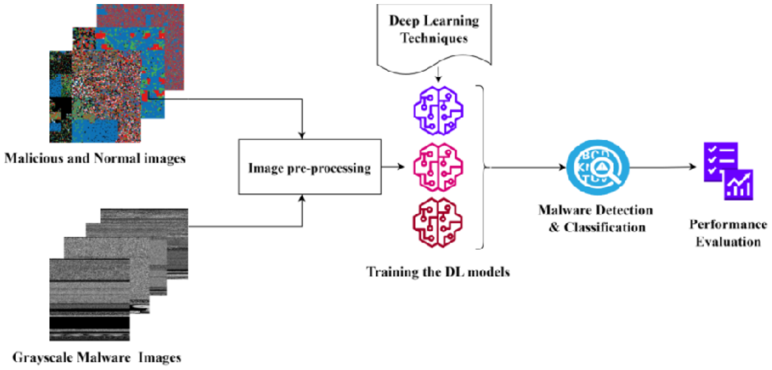


Figure 1: Proposed Methodology

Benchmark Datasets

In ML and DL techniques, the dataset is crucial for performance. Here, two datasets are used: (1) **Dataset 1**, the original grayscale image dataset Malimg[11], comprising 3,993 image samples across 25 classes for multiclass classification; and (2) **Dataset 2**, which consists of 48,240 malware samples along with a visualized image dataset for anomaly detection [25], containing 24,109 samples of both malicious and benign visualized images.

Image Pre-Processing

In the 1D-CNN approach, the image size is reduced to 32×32 width and height and transformed into a one-dimensional array of length 1024, which is then saved in a separate CSV file format. For the transfer learning approach, we utilized pre-trained models and resized the Malimg dataset images to a uniform size of 224×224 pixels. The dataset 2, consists the malware and normal images of size 256×256, and 3 channels of RGB. The images are resized to input image size of 224×224.

Deep Learning Techniques

There is always a trade-off between the accuracy and efficiency of the various DL techniques. The more complex models require high computational resources and time. The we assessed the performance of models utilizing 1D-CNN, transfer learning, and ensemble techniques.

1D-CNN Technique

The 1D-CNN approach is used where we have the input dataset in one dimensional format.Here, the malware images are represented as pixel values stored in a one-dimensional array within a CSV file. This dataset is utilized for training and testing the models.

Transfer Learning Technique

The transfer learning approach saves the time of feature learning, instead it uses the past learning experience from the large dataset. The transfer learning models are fine tuned for the task.

Ensemble Technique

The ensemble technique allows to use the prediction of multiple ML models to make the optimized prediction. Here we have used the stacked ensemble techniques in two different methods: (1) Average ensemble technique and (2) Weighted average ensemble technique.

The integration of transfer learning and ensemble learning techniques is essential for enhancing the accuracy, efficiency, and robustness of malware classification systems, providing vital protection against increasingly sophisticated cyber threats.

Performance Metrics

In ML, particularly for classification tasks such as malware detection and intrusion detection, various performance metrics are utilized to assess a model's effectiveness. the following metrics is used for performance analysis.

Accuracy

Accuracy measures the overall correctness of the model by calculating the proportion of correctly predicted instances including both positive and negative out of all predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

Precision is also called positive predictive value. It measures the proportion of true positives out of all the positive predictions made by model.

$$Precision = \frac{TP}{TP + FP}$$

Recall

Recall is also called sensitivity or true positive rate. It measures the proportion of true positives out of all actual positive cases.

$$Recall = \frac{TP}{TP + FN}$$

ROC-AUC

Useful for evaluating the model's ability to distinguish between classes across all thresholds. It plots the ROC (Receiver Operating Characteristic) curve and compute the AUC (Area Under the Curve) score range of 0 to 1.

Confusion Matrix

Provides detailed insights into the classification errors made by the model, useful for multi-class problems.A confusion matrix is a table used to evaluate the performance of a classification model. It provides a summary of prediction results on a classification problem by comparing actual target values with predicted values.

Experimental Environment

The experimental setup encompasses the environmental configuration necessary to meet the hardware and software requirements, dataset split and hyperparameter tuning, as detailed below.

Environment Setup

The experimental environment utilized is Google Collaboratory, which features Python 3 and an A100 GPU, equipped with 83.5 GB of system RAM, 40 GB of GPU RAM, and 201.3 GB of disk storage. This virtual computing platform has installed Python (version 3.10.12), Tensor Flow (version 2.15.0), and the NumPy library.

Dataset Split

The dataset is divided as 80% for training and 20% for testing. For dataset-1, which consists of a total of 24,109 visualized malware and normal image samples, 19,288 samples are allocated for training, while 4,821 samples are reserved for testing and validation. The dataset 2 is divided into a training set of 5,976 samples, a validation set of 1,495 samples, and an additional test set containing 1,868 malware samples.

Hyperparameter Tuning

Hyperparameters are critical parameters used to fine-tune the model for optimal performance. In this study, the activation function applied to the internal layers is 'ReLU,' while the final layer employs 'softmax' for both binary and multiclass classification. For the loss function, 'categorical_crossentropy' is utilized for multiclass classification, whereas 'binary_crossentropy' is used for binary classification tasks. A learning rate of 0.001 was found to be effective for training the model. Additionally, a batch size of 32 is employed, with the Adam optimizer selected for optimization.

EXPERIMENTAL ANALYSIS AND RESULTS

In the experimental phase, we developed deep learning models utilizing 1D-CNN, transfer learning, and ensemble learning techniques. These models were trained and tested on two distinct benchmark datasets, and their performance was rigorously evaluated.

Malware Classification

The malware classification task was conducted for a multiclass classification model. We performed two experimental setups based on the dataset type: one utilizing the same dataset in CSV format, where image pixels are stored in a one-dimensional format, and the other using a 2D grayscale image dataset. In both cases, we employed the Malimg dataset (dataset-1), which consists of 3,993 image samples across 25 classes. For the CSV image dataset, we applied a 1D-CNN approach, using images resized to 32×32 pixels and converted into a 1D array of 1,024 pixels.To classify malware images, we used a custom 1D-CNN model CNN2+LSTM[13] and modified the classical 1D-VGG16 model. Both models were trained on a dataset of 7,471 malware image samples over 200 epochs (Figure 2), achieving classification accuracies of 98.12% and 97.64% on the test dataset, respectively.

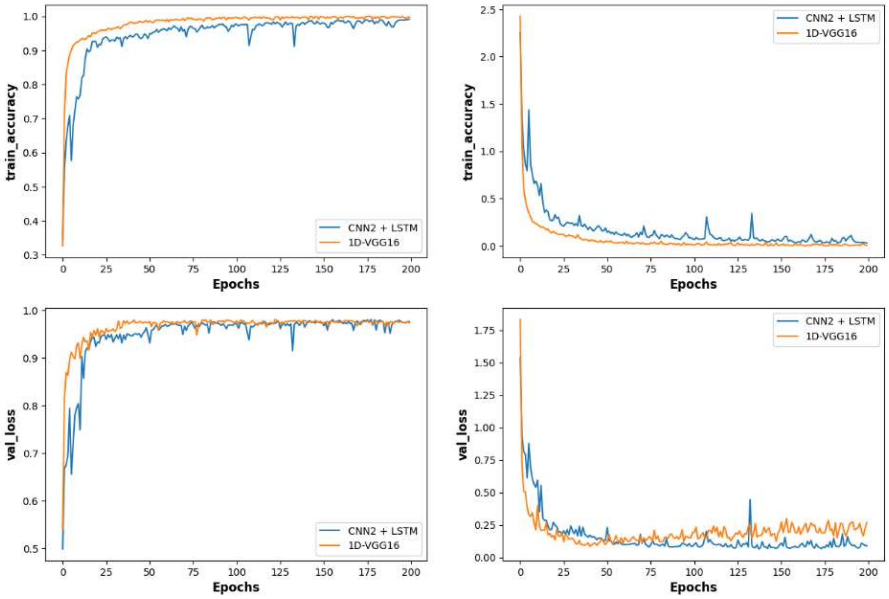


Figure 2: Training and Validation Accuracy Loss Curve for Malware Classification.

The Table I shows the test accuracy of models on the validation dataset and weighted values of precision, recall, and F1-score. The CNN2+LSTM model achieves higher accuracy.

Table 1: Performance of Malware Classification Models

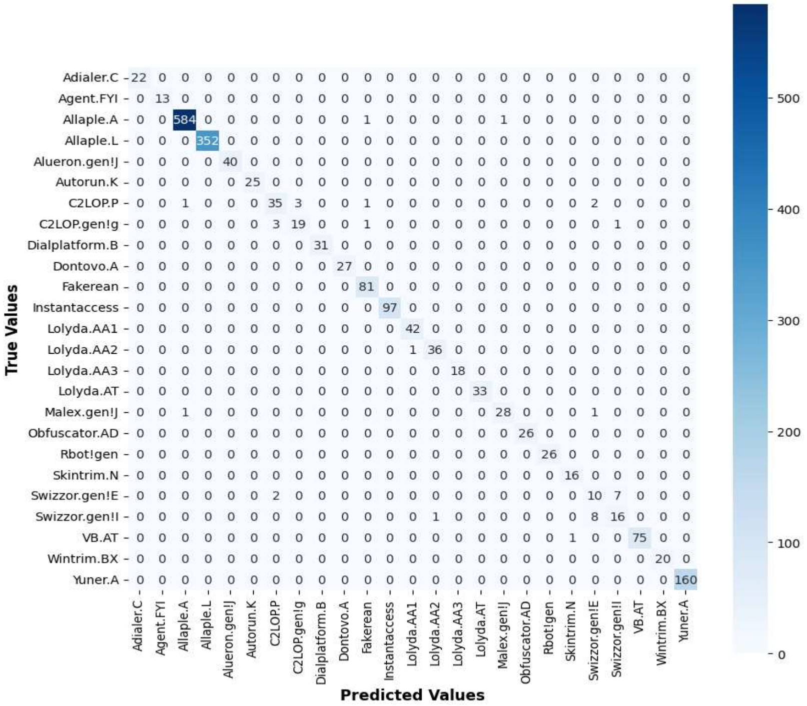
Proposed Models (1D-CNN)	Accuracy	Precision	Recall	F1-score
CNN2+LSTM	98.12%	0.98	0.98	0.98
1D-VGG16	97.64%	0.98	0.98	0.98

After the evaluating the 1D-CNN models, the ensemble model is created using the two methods: Average ensemble method, and Weighted average ensemble method. Twoensemble models that integrated the highest-performing models from both the custom and classical categories. The top-performing models were then combined into two ensemble models, designated as Ensemble Model 1 and Ensemble Model 2, utilizing a stacking ensemble learning method to leverage multiple base models for enhanced performance. Here, the average ensemble method is used to combine the two 1D-CNN models and make an ensemble model.

The Table II shows the test accuracy of ensemble models and weighted precision, recall, and F1-score values for the models. The ensemble model 2 enhanced the test accuracy up to 98.44%.

Table 2: Performance of Malware Classification using Ensemble Techniques

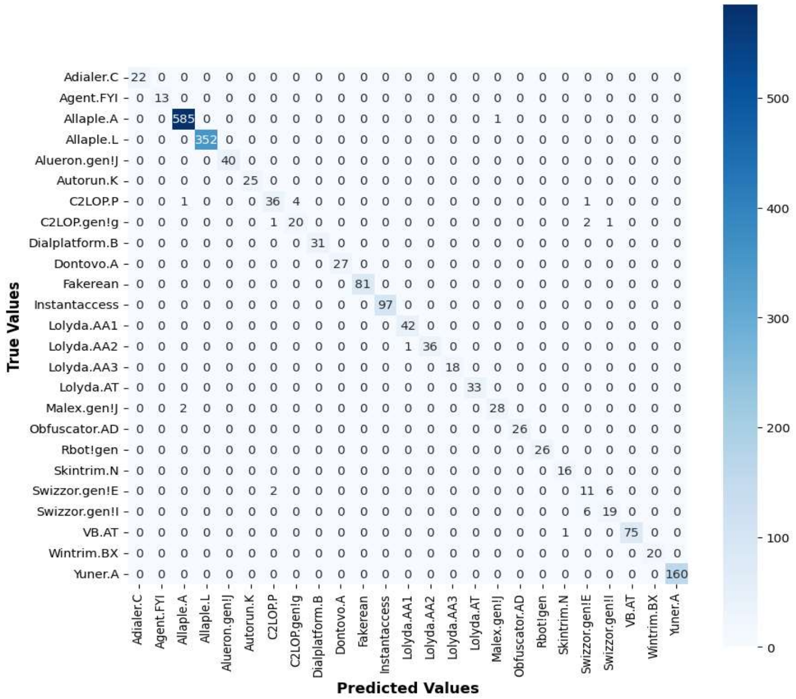
Proposed Models (1D-CNN)	Accuracy	Precision	Recall	F1-score
Ensemble model 1	98.07%	0.98	0.98	0.98
Ensemble model 2	98.44%	0.99	0.99	0.99



	Adialer.C	Agent.FYI	Allaple.A	Allaple.L	Alueron.gen!J	Autorun.K	C2LOP.P	C2LOP.gen!g	Dialplatform.B	Dontovo.A	Fakerean	Instantaccess	Lolyda.AA1	Lolyda.AA2	Lolyda.AA3	Lolyda.AT	Malex.gen!J	Obfuscator.AD	Rbot!gen	Skintrim.N	Swizzor.gen!E	Swizzor.gen!I	VB.AT	Wintrim.BX	Yuner.A
Adialer.C	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Agent.FYI	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Allaple.A	0	0	584	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Allaple.L	0	0	0	352	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alueron.gen!J	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Autorun.K	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2LOP.P	0	0	1	0	0	0	35	3	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0
C2LOP.gen!g	0	0	0	0	0	0	3	19	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Dialplatform.B	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dontovo.A	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fakerean	0	0	0	0	0	0	0	0	0	0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Instantaccess	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA1	0	0	0	0	0	0	0	0	0	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA2	0	0	0	0	0	0	0	0	0	0	0	0	1	36	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0
Lolyda.AT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0
Malex.gen!J	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	1	0	0	0	0
Obfuscator.AD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0
Rbot!gen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0
Skintrim.N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0
Swizzor.gen!E	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	10	7	0	0	0
Swizzor.gen!I	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	8	16	0	0	0
VB.AT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	75	0	0
Wintrim.BX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0
Yuner.A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160

Figure 3: Confusion Matrix of Ensemble Model 1.

Figure 3 and Figure 4 show the confusion matrices for both ensemble models and the classification accuracy for eachclass and the misclassified image samples.



True Values	Adialer.C	Agent.FYI	Allaple.A	Allaple.L	Alueron.gen!J	Autorun.K	C2LOP.P	C2LOP.gen!g	Dialplatform.B	Dontovo.A	Fakerean	Instantaccess	Lolyda.AA1	Lolyda.AA2	Lolyda.AA3	Lolyda.AT	Malex.gen!J	Obfuscator.AD	Rbot!gen	Skintrim.N	Swizzor.gen!E	Swizzor.gen!I	VB.AT	Wintrim.BX	Yuner.A
Adialer.C	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Agent.FYI	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Allaple.A	0	0	585	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Allaple.L	0	0	0	352	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Alueron.gen!J	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Autorun.K	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C2LOP.P	0	0	1	0	0	0	36	4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C2LOP.gen!g	0	0	0	0	0	0	1	20	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0
Dialplatform.B	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dontovo.A	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fakerean	0	0	0	0	0	0	0	0	0	0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Instantaccess	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA1	0	0	0	0	0	0	0	0	0	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA2	0	0	0	0	0	0	0	0	0	0	0	0	1	36	0	0	0	0	0	0	0	0	0	0	0
Lolyda.AA3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0
Lolyda.AT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0
Malex.gen!J	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0
Obfuscator.AD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0
Rbot!gen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0
Skintrim.N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0
Swizzor.gen!E	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	11	6	0	0	0
Swizzor.gen!I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	19	0	0	0
VB.AT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	75	0	0
Wintrim.BX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0
Yuner.A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160

Figure 4: Confusion Matrix of Ensemble Model 2.

Malware Detection

For malware detection, the transfer learning-based Efficient Net models are selected. These models were trained on a dataset of 19,288 binary class samples over 30 epochs. The performance of the models was evaluated on a binary class dataset comprising 24,019 images of malicious and normal classes (dataset 2). Figure 5 illustrates the training and validation accuracy loss curve for the transfer learning-based Efficient Net models, highlighting that the EfficientNetV2B0 model outperforms the others.

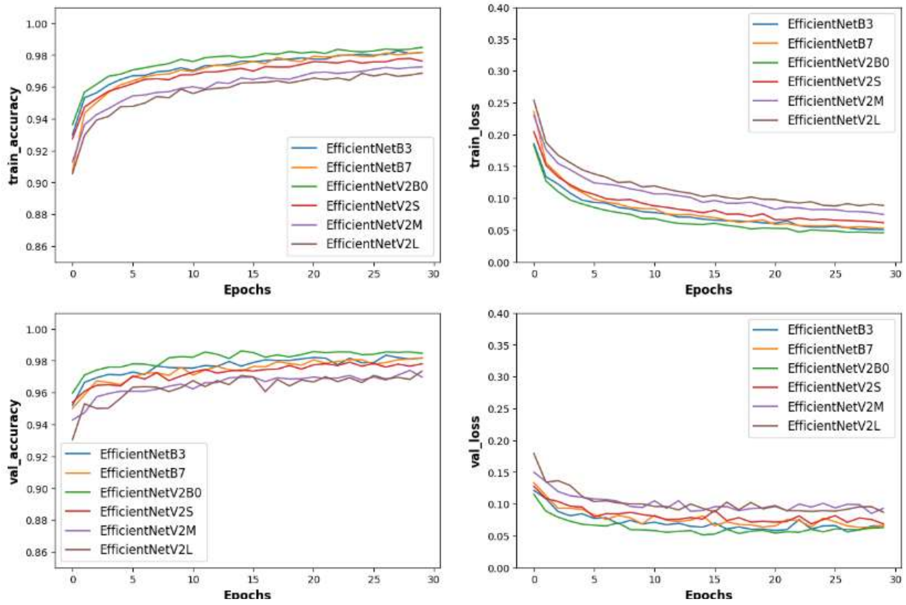


Figure 5: Training and Validation Accuracy Loss Curve for Malware Detection.

The performance of six Efficient Net models (EfficientNetB3, EfficientNetB7, EfficientNetV2B0, EfficientNetV2S, EfficientNetV2M, and EfficientNetV2L) is trained on dataset 2, and their performance is evaluated. The top-performing models EfficientNetB3, EfficientNetV2B0, and EfficientNetB7 with validation accuracy of 98.15%, 98.47%, and 98.17%, respectively, are selected to create an ensemble model aimed at enhancing accuracy. Here, two ensemble models— (1) average ensemble model and (2) weighted average ensemble modelwere created. The average ensemble model achieved a test accuracy of 98.83%, while the weighted average ensemble model achieved a test accuracy of 98.71%. Figures 6 and 7 display the confusion matrices for the average and weighted average ensemble models based on the total test dataset of 4,821 samples from the binary classes of dataset 2.

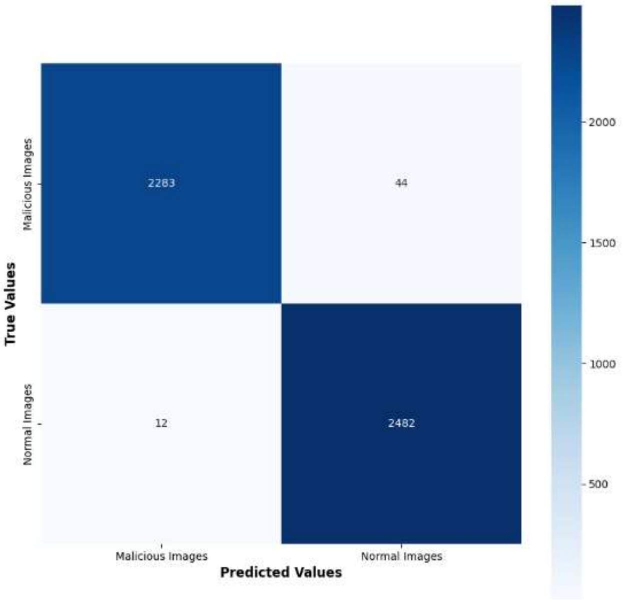


Figure 6: Confusion Matrix of Average Ensemble Model.

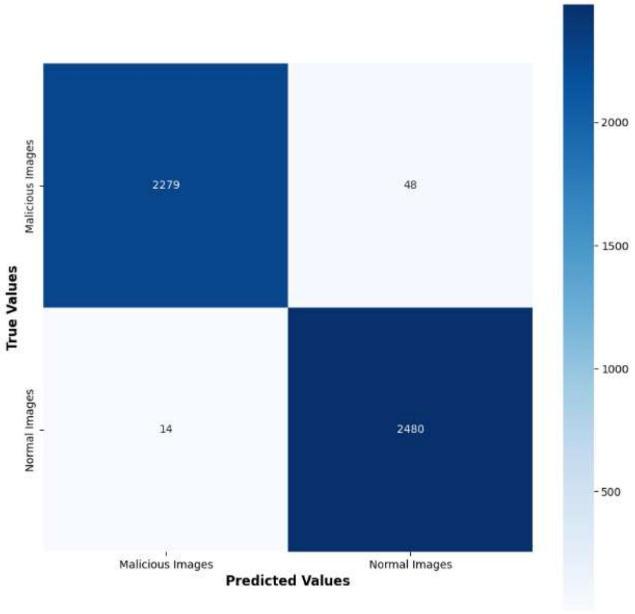


Figure 7: Confusion Matrix of Weighted Average Ensemble Model.

The performance of average and weighted average ensemble models is given in Table III, along with the weighted precision, recall, and F1-score of the models. Here, the average ensemble model achieves the higher malware detection accuracy of 98.83%.

Table 3: Performance of Malware Detection Models

Proposed Models	Accuracy	Precision	Recall	F1-score
Average ensemble model	98.83%	0.99	0.99	0.99
Weighted average ensemble model	98.71%	0.99	0.99	0.99

RESULTS AND DISCUSSION

From the experimental outcome of applied techniques, the three best results from each experimental work are summarized in Table IV. The transfer learning approach gives the highest validation accuracy but also comes with the cost of higher training time. The 1D-CNN model employing the ensemble techniques takes minimum time to train and gives acceptable classification accuracy. The accuracy of the 1D-CNN model is significantly enhanced when the ensemble technique is applied. Additionally, the combination of transfer learning, which utilizes 2D input images of size 224×224, with the ensemble approach further elevates the overall performance.

Table 4: Performance Comparision of DL Techniques

Deep Learning Techniques	Dataset	Number of Classes	Training Time	No. of Epochs	Training accuracy	Validation accuracy
1D-CNN (CNN2 + LSTM)	Dataset-1	25	10 min 12 s	200	99.23%	98.12%
1D-CNN + Ensemble	Dataset-1	25	02 min 07 s	10	99.95%	98.44%
Transfer learning + Ensemble	Dataset-2	02	1h 3 min	30	99.48%	98.83%

The corresponding ROC-AUC curve for the best performing models is displayed in Figure 8. The value of area under curve is maximum for the 1D-CNN ensemble model for the multiclass classification.

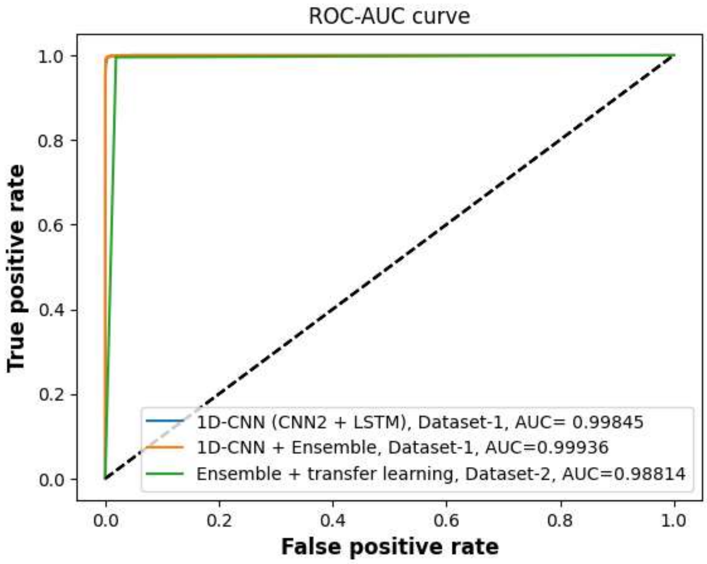


Figure 8: ROC-AUC Curve for Best Performing Models.

Performance Comparison with State-of-the-Art

The best-performing models are also compared with the state-of-the-art with respect to the Malimg dataset. The comparison is shown in Table V.

Table 5: Comparison with State of the Art

References	Architecture Or Model	Dataset	Accuracy (%)
[11]	GIST, KNN	Malimg	97.18
[13]	1D-CNN, LSTM	Malimg	96.3
[14]	ResNeXt50	Malimg	98.86
[15]	ResNet50, AlexNet, Inception-v3	Malimg	97.78
[16]	VGG19	Malimg	97.68
[22]	VisMal	Malimg	96
[26]	IMCBL	Malimg	97.64
[27]	SODCNN-IMC	Malimg	98.42
(Proposed)	1D-CNN (CNN2 + LSTM)	Malimg (dataset 1)	98.12%
(Proposed)	1D-CNN + Ensemble	Malimg (dataset 1)	98.44%
(Proposed)	Transfer learning + Ensemble	Dataset 2	98.83%

CONCLUSION AND FUTURE SCOPE

Through the deep learning approach, which incorporates 1D-CNN, transfer learning, and ensemble techniques, the performance of various models is evaluated on the benchmark datasets. The experimental findings indicate that visualization-based malware detection and classification effectively leverage computer vision techniques, particularly convolutional neural networks. The 1D-CNN model uses a smaller number of resources and time to train the models as compared to the 2D-CNN model. The transfer learning models are much more efficient to train on the RGB image dataset by using its pre-trained values of weights. The ensemble techniques further enhanced the model's performance. The ensemble technique significantly enhanced model accuracy, achieving a peak malware detection accuracy of 98.83% and malware classification accuracy of 98.44%. In the current cybersecurity landscape, there is a pressing need for updated and novel malware datasets to effectively detect emerging types of malwares. The future scope includes the efficient use of CNN, transfer learning, generative adversarial networks (GAN), and ensemble techniques for visualization of network data packets and intrusion detection, as well as to address other cyber threats such as phishing and ransomware detection. The real-time malware analysis is crucial for protecting the security and stability of networks, systems, and sensitive data across various industries. From enterprise and government sectors to IoT environments and healthcare systems, these technologies offer a robust shield against increasingly sophisticated cyber threats.

REFERENCE

1. *K. Kumar and A. Dwivedi, "Big Data Issues and Challenges in 21 st Century," International Journal on Emerging Technologies (Special Issue NCETST-2017), vol. 8, no. 1, pp. 72–77, 2017, [Online]. Available: www.researchtrend.net*
2. *H. L. M. A. N. Prachi Chauhan, "A Review: Security and Privacy DefensivTechniques for Cyber Security Using Deep Neural Networks (DNNs)," in Advanced Smart Computing Technologies in Cybersecurity and Forensics, I., 2021, pp. 11–22.*

3. *S. M. Rao and A. Jain, "Advances in Malware Analysis and Detection in Cloud Computing Environments: A Review," International Journal of Safety and Security Engineering, vol. 14, no. 1, 2024, doi: 10.18280/ijsse.140122.*
4. *A. Aliahmad, D. Eleyan, A. Eleyan, T. Bejaoui, M. F. Zolkipli, and M. Al-Khalidi, "Malware Detection Issues, Future Trends and Challenges: A Survey," in 2023 International Symposium on Networks, Computers and Communications, ISNCC 2023, 2023. doi: 10.1109/ISNCC58260.2023.10323624.*
5. *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," 2022. doi: 10.3390/app12178482.*
6. *M. Rai and H. L. Mandoria, "A Study on Cyber Crimes, Cyber Criminals and Major Security Breaches," International Research Journal of Engineering and Technology, no. July, 2008.*
7. *A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0038-7.*
8. *A. Dwivedi, R. P. Pant, S. Pandey, and K. Kumar, "Internet of Things' (IoT's) Impact on Decision Oriented Applications of Big Data Sentiment Analysis," in Proceedings - 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018, 2018. doi: 10.1109/IoT-SIU.2018.8519922.*
9. *N. Marastoni, R. Giacobazzi, and M. Dalla Preda, "Data augmentation and transfer learning to classify malware images in a deep learning context," Journal of Computer Virology and Hacking Techniques, vol. 17, no. 4, 2021, doi: 10.1007/s11416-021-00381-3.*
10. *V. Priya and A. S. Sofia, "Review on Malware Classification and Malware Detection Using Transfer Learning Approach," in Proceedings - 5th International Conference on Smart Systems and Inventive Technology, ICSSIT 2023, 2023. doi: 10.1109/ICSSIT55814.2023.10061076.*
11. *L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in ACM International Conference Proceeding Series, 2011. doi: 10.1145/2016904.2016908.*
12. *A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," Int J Comput Vis, vol. 42, no. 3, 2001, doi: 10.1023/A:1011139631724.*
13. *R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2906934.*
14. *J. H. Go, T. Jan, M. Mohanty, O. P. Patel, D. Puthal, and M. Prasad, "Visualization Approach for Malware Classification with ResNeXt," in 2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings, 2020. doi: 10.1109/CEC48606.2020.9185490.*
15. *O. Aslan and A. A. Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3089586.*
16. *M. J. Awan et al., "Image-based malware classification using vgg19 network and spatial convolutional attention," Electronics (Switzerland), vol. 10, no. 19, 2021, doi: 10.3390/electronics10192444.*

17. S. O'Shaughnessy and S. Sheridan, "Image-based malware classification hybrid framework based on space-filling curves," ComputSecur, vol. 116, 2022, doi: 10.1016/j.cose.2022.102660.

18. Q. Lu, H. Zhang, H. Kinawi, and D. Niu, "Self-Attentive Models for Real-Time Malware Classification," IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3202952.

19. S. Seneviratne, R. Shariffdeen, S. Rasnayaka, and N. Kasthuriarachchi, "Self-Supervised Vision Transformers for Malware Detection," IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3206445.

20. A. Vaswani et al., "An image is worth 16*16 words: transformers for image recognition at scale," in Advances in Neural Information Processing Systems, 2017.

21. S. Freitas, R. Duggal, and D. H. Chau, "MalNet: A Large-Scale Image Database of Malicious Software," in International Conference on Information and Knowledge Management, Proceedings, 2022. doi: 10.1145/3511808.3557533.

22. F. Zhong, Z. Chen, M. Xu, G. Zhang, D. Yu, and X. Cheng, "Malware-on-the-Brain: Illuminating Malware Byte Codes With Images for Malware Classification," IEEE Transactions on Computers, vol. 72, no. 2, 2023, doi: 10.1109/TC.2022.3160357.

23. R. El-Sayed, A. El-Ghamry, T. Gaber, and A. E. Hassanien, "Zero-Day Malware Classification Using Deep Features with Support Vector Machines," in Proceedings - 2021 IEEE 10th International Conference on Intelligent Computing and Information Systems, ICICIS 2021, 2021. doi: 10.1109/ICICIS52592.2021.9694256.

24. H. Al-Qadasi, D. Y. M. Benchadi, S. Chehida, K. Fukui, and S. Bensalem, "Neural Network Innovations in Image-Based Malware Classification: A Comparative Study," 2024, pp. 252–265. doi: 10.1007/978-3-031-57916-5_22.

25. B. Saridou, J. Rose, S. Shiaeles, and B. Papadopoulos, "48,240 Malware samples and binary visualisation images for machine learning anomaly detection (2021)," 2022.

26. D. Vasan, M. Hammoudeh, and M. Alazab, "Broad learning: A GPU-free image-based malware classification," Appl Soft Comput, vol. 154, 2024, doi: 10.1016/j.asoc.2024.111401.

27. S. Duraibi, "2024 IEEE Enhanced image based malware classification using snake optimization algorithm with deep cnn," IEEE Access, 2024.