Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Enhanced Malware Detection for IoT Networks Utilizing a 2D-CNN with Data Augmentation on the Malimg Dataset

Iman Fadhil Saleh

Ministry of Education, Kirkuk, Iraq. Email: emanibo399@gmail.com

ARTICLE INFO

Article history:
Received: 26 /1/2025
Rrevised form: 9 /2/2025
Accepted : 2 /3/2025
Available online: 30 /3/2025

Keywords:

Internet of Things (IoT),
Convolutional Neural Network ,
Deep learning, Cybersecurity,
Malware detection.

ABSTRACT

With the swift growth of the Internet of Things (IoT), the attacker's threat surface has increased multi-fold, making IoT networks a hotbed for multiple malware types. IoT networks' ever-changing and diverse structures make it almost impossible for traditional malware detection systems to effectively identify and classify hostile traffic in real-time. One viable approach to solving the issue is training Convolutional Neural Networks (CNNs) with data augmentation techniques, as expanding the dataset could help improve IoT malware detection by enabling better classification of distinct malware traffic patterns. In order to enhance detection performance, the authors of this paper suggest a unique two-dimensional (2D) CNN architecture that is tailored for the Malimg dataset and incorporates data augmentation techniques. The suggested method offers a major improvement over other studies that used manually designed dataset-specific characteristics for malware classification by automatically extracting features straight from the infected pictures. By minimizing overfitting, this technique allows the model to train efficiently over a mere ten epochs. The model is better able to generalize and adjust to a greater range of malware samples by employing data augmentation. The outcome of the model built on the Single-CNN architecture performs better than other established models like DenseNet, VGG16, and the VBDN framework obtained an accuracy of 98.86%. The paper included comparative graphs and line plots, bar charts and pie charts demonstrating effectiveness of the proposed model and its original version. These outcomes emphasize the value of domain specific optimizations to solve intricate problems of malware detection and cybersecurity. These results put attention to sophisticated techniques needed to solve malware detection problems in IoT networks and underscore the need for evolving methodologies for these emerging issues.

BSC.

https:://doi..org/10.29304/jqcsm.2025.17.11973

1. Introduction

In the present era, the Internet has become a vital component of daily life and a major means of exchanging information. Cybercriminals are launching cyberattacks on Internet-connected services because of their open public availability and low user awareness. Malware threat detection and family categorization is becoming a complicated issue among other intrusions[1]. Malware assaults are a serious security risk to cyberspace because of their

∗ Iman Fadhil Saleh

Email addresses: emanibo399@gmail.com

Communicated by 'sub etitor'

evolving nature. Malware is a type of code that deviates from the standard flow of a computer's activities. Viruses, trojans, worms, botnets, ransomware, downloaders, information thieves, rootkits, and other malware threats vary in their functions[2] . These days, cybersecurity experts face many difficulties due to the exponential expansion and evolution of malware [3] . Deep learning (DL) has become a powerful tool, offering excellent advantages in many fields and applications. It has been utilized in different study fields because of its fast development[4]. Obfuscation, polymorphism, metamorphism, and encryption methods are employed by attackers to change and conceal the form of such malicious software. As a result, traditional approaches like heuristic-based and signature-based analysis are no longer effective, necessitating the development of more advanced analytic algorithms that can identify the patterns and behavior of malware. As deep learning has developed, methods like Convolutional Neural Networks [5] (CNN) have emerged as strong mechanisms for identifying intricate patterns within massive picture datasets. An CNN creates a series of feature maps while scanning an image, showing where the features that each filter has looked for are on the image. The authors in [6] have suggested an approach based on textural features of images to convert binary code of malware families into image format. This method uses Convolutional Neural Networks (CNNs), which utilize deep learning techniques, to identify malware through the conversion of binary code into RGB or grayscale images. This study banks on the CNNs' capability to identify patterns within the datasets formed from such images and understand the overall code structure. In [7] used the method of converting binary code into a picture, which identified and categorized malware samples with 98.41% accuracy and 0.08078 log loss. The current study builds on by investigating the use of hyperparameters and other strategies to improve the effectiveness of malware classification performance metrics using CNNs across three benchmarking datasets and a new malware family dataset, as well as the efficiency of model training execution time. Datasets used for malware analysis are frequently unbalanced, having a greater number of samples in one class than in others. Class imbalance was examined in this study using three popular datasets: the MaleVis dataset [8], the Malimg dataset [5], and the Microsoft Big 2015 datasets [9].

Furthermore, we presented the Fusion dataset, which has 32,601 samples from 59 malware families and merges the first three datasets. The Fusion Dataset serves as a standard for assessing the resilience of CNN classification systems in addition to offering a wide and varied representation of malware types. The Internet of Things (IoT) has recently made it possible for individuals to connect with their homes and workplaces without the need for human-computer contact because to advancements in information and communication technology (ICT) [10], [11]. Through process automation in businesses and industries, the Internet of Things ecosystem [12] has made it possible for objects (things) including computers, servers, people, digital machines, and networks—which are utilized almost everywhere in the world—to lower labor costs and expenses[13] . In order to precisely identify metamorphic malware in IoT devices, we employed a deep learning model in this research. Hospitals, enterprises, households, and organizations [14] have been able to continually embrace interconnected IoT devices for greater connection and convenience due to the breadth of IoT applications. Traditional computer platforms use sensors or hardware that has a defined purpose. IoT technology must, however, instantly give decision makers entity-level maintenance, logistical, and intelligence data so they may respond more quickly and confidently. Many IoT devices can now connect and access the Internet fast thanks to advancements in ICT technology. Owing to the Internet's openness, a number of well-publicized risks and assaults are highlighting these interconnected IoT devices' susceptibility [12]. As IoT devices have proliferated, the security problem has become worse[15]. For instance, in 2019 there were an average of 5200 assaults per month against IoT devices [16]. Malware assaults pose a hazard to these network systems and interconnected devices over the Internet worldwide, among other cyber security concerns and problems [17]. To enhance IoT malware detection, this research proposes an innovative 2D CNN model specifically designed for the Malimg dataset with augmentation. By feature extraction from malware images, the model's overfitting issues are solved and the need for manually crafted features is avoided. The proposed Single-CNN architecture achieved a 98.86% accuracy, outperforming other models like DenseNet, VGG16, and VBDN. This research showed a number of comparison visualizations to illustrate the efficacy of the proposed approach and highlighted the importance of domain specific optimization to solve the IoT malware detection problem.

2. Related Work

The authors of [18] suggested using a reweighted class-balanced loss function in the first classification layer of their DenseNet-based model. The goal of this strategy was to increase malware classification performance. For the Microsoft BIG 2015 dataset, the accuracy is 98.23 %; for the Malimg dataset, it is 98.23%; for the Male-Vis dataset, it is 98.21%; and for the unseen Malicia dataset, it is 89.48%. Pachhala et al. model, [19] proposed balance-augmented VGG16 model and used that to improve the classification accuracy for imbalanced datasets. They

highlight the need for countering data imbalance, a prevalent problem with malware datasets, by using augmentation methods that maintain equal presence of malware families. With this, the model performance saw great improvements showing that merging classical CNN architecture with domain-specific preprocessing strategies held promising potential. The mechanism they devise thus serves as an important foundation for dealing with data imbalance in image-based malware detection.

A new strategy [20] In IoT devices, ANNs based on machine learning algorithms are used to identify Benign and Mirai. Matlab2018b is implemented and trained using the Mirai and Benign datasets. The results collected demonstrate a notable improvement in malware detection accuracy and false-negative rates in IoT systems.

An effective and versatile convolutional neural network (CNN)-based approach for multi-class malware detection was presented by Liu et al. [21]. Their technique, in contrast to conventional signature-based methods, converts malware binaries into pictures so that CNNs may be used to extract and categorize characteristics. The authors showed that their algorithm works better than current methods in terms of accuracy and computing efficiency, especially when dealing with a variety of malware variants. This study demonstrates the potential of DL models based on images in cybersecurity applications and offers a starting point for investigating additional improvements, including lightweight structures for real-time detection.

[22] suggested a hybrid method for malware detection and classification that combines ML and DL techniques. They employed both local and global virus image characteristics as well as hybrid textural elements. While a CNN model was trained using the global feature, machine learning classifiers were trained using the local features.

Their preprint study [23] Proposed malware detection and classification via transfer learning. The authors compared the classical machine learning k-NN method and deep pre-trained ResNet34 model by using the Malicia dataset. The authors presented the deep pre-trained model, i.e., ResNet34 outperformed the classical k-NN model regarding performance accuracy. However, they have not compared their approach with the existing deep learning models. Also, the authors have used imbalanced datasets of 9895 malware and 704 benign samples, which seems biased for malware classes in training the model. They did not address the data imbalance problem in their research.

For an improved performance of network traffic detection, Riyaz and Ganapathy [24] adopted the conditional random fields and linear correlation coefficients method of feature selection with the purpose of recognizing the most important feature attributes. Further, a CNN model was used for feature extraction. In the works of Azizjon et al. [25] it was shown that the 1D-CNN model of supervised learning for network traffic temporal patterns performs better than random forest and support vector machine models.

In order to extract the temporal and geographical aspects of traffic roadways and provide more precise forecasts of road traffic flow. [26] suggested combining a graph convolutional network with a Gated Recurrent Unit (GRU). According to the results, it performs better than more conventional time series regression models like SVR and ARIMA.

Table 1 Summarization of the Related work

Study	Datasets	Results	Strengths	Limitations
[18] DenseNet-centered model	Microsoft GIANT 2015, Malimg, Male-Vis, Malicia	Microsoft MAJOR 2015: 98.23%, Malimg: 98.23%, Male-Vis: 98.21%, Malicia: 89.48%	Reweighted class-consistance loss ceremony to adjust division performance.	No straightforward conversation on how the model handles data imbalance, balanced regardlessly the emphasis is on bettering sincerity.
[19] Balance-upgraded VGG16	Imbalanced malware datasets	Finer ordering truthfulness for imbalanced	Combines ancient CNN design with domain-unique	Aim on data intensification without directly

		datasets.	preprocessing schemes to tackle data imbalance.	addressing different challenges in deep understanding for malware detection.
[20] ANNs for IoT	Mirai and Peaceful datasets	Influential development in malware detection credibility and reduced false-negative rates.	Focuses on boosting IoT-targeted malware detection with machine studying.	Limited comparison with additional strategies and no precise performance evaluation across multiple datasets.
[21] CNN-emerged multi-class malware detection	Malware binary datasets	Outperformed classical signature-built tactics in authenticity and analyzing optimization.	Converts malware binaries into images for CNN-relying feature extraction, offering more suitable genuineness.	The path may become limited by the scalability of CNNs for trustworthy-time or asset-constrained environments.
[22] Hybrid ML and DL model	Virus image datasets	CNN-built global feature extraction mixed with machine studying classifiers for local traits.	Hybrid method combines CNN with machine learning, bettering detection validity.	Limited dataset diversity and ability for model overfitting due to complexity.
[23] Transfer learning with ResNet34	Malicia dataset	ResNet34 outperformed k-NN in performance genuineness.	Uses transfer understanding with a pre-trained model to correct performance.	Imbalanced dataset used, best to potentiality prejudice in training. No comparison with extra deep understanding models.
[24] Conditional random fields with CNN	Network traffic datasets	Improved network traffic detection performance.	Conditional random fields associated with CNNs help recognize essential feature attributes.	Lacks direct center on malware detection and doesn't address exact malware sorting challenges.
[25] 1D-CNN model for network traffic	Network traffic datasets	1D-CNN outperformed random forest and SVM in network traffic evaluation.	Powerful use of 1D-CNN for supervised understanding of network traffic temporal patterns.	Limited emphasis on malware detection in network traffic and not as usable to image-emerged malware detection.
[26] Graph convolutional network and GRU	Road traffic datasets	Outperformed customary moment-series models like SVR and ARIMA for traffic flow prediction.	Uses a combination of graph convolutional network and GRU for precise traffic flow estimating.	Not concentrated on malware detection, and primarily targeted at road traffic anticipation rather than system security.

3. Methodology

Unusual traffic in Internet of Things networks can significantly impact device performance and network operations. Detecting traffic anomalies is challenging due to the vast range of characteristics of aberrant traffic, including changes in packet size, frequency, and timing [27]. To address this issue, machine learning (ML)-based image classification methods are proposed, which can be trained to recognize specific characteristics or anomalies in traffic patterns represented as pixels or vectors in an image. Conventional machine learning models like SVM, decision trees, and random forests can be used to classify traffic patterns. However, human extraction of pertinent characteristics from photos is a challenge, and feature extraction becomes increasingly complex as the number of traffic anomaly types increases[28]. Convolutional Neural Networks (CNN), a deep learning technique, are used to automatically extract important and educational components for each type of aberrant traffic. CNNs have achieved cutting-edge outcomes in computer vision and have become the go-to method for picture categorization problems. [29]. CNNs in IoT networks provide a more efficient way by automatically identifying a myriad of patterns in the traffic data associated with the most unknown types. However, it might be tricky to detect patterns with ML models in low power or hidden aberrant traffic, since noise might hide the signal. Consequently, to tackle this limitation, the classification procedure takes the frequency and temporal domains and puts more supplementary information into it [27]. The Feature-Aided CNN classifier improves detection performance in IoT networks by tying deep learning techniques to a signal's characteristics. Therefore, this study looks at how well it can detect anomalous traffic and how it might improve the resilience of IoT networks. Because of its shift invariance, CNNs are widely employed for image identification and classification. This allows them to be utilized for handwritten character and human face recognition [30][31]. They are comprised of at least one convolutional layer bound to a dense layer, which helps in condensing the initial images to a smaller feature set by minimizing the number of weights needed. CNN consist of:

Convolutional Layer: The CNN input format is covered first in this section. CNN takes in a multi-channeled image, whereas other neural networks employ vector formats. For instance, the RGB image format includes three channels, but the grayscale picture format only has one. Look at the following: To learn more about the convolutional process, use a 2 x 2 random weight-initialized kernel on a 4 × 4 grayscale image. The kernel first scans the whole picture in both horizontal and vertical directions. The input image and the kernel's dot product are also calculated, which involves adding and multiplying each of their distinct values to produce a single scalar result. The procedure is then carried out again until slipping is impossible. The outcome is represented by the calculated dot product values. Figure 1 clearly illustrates the fundamental computations made at each step. The 2 2 kernel is represented in this example by the light green hue, while a section of the same-sized input image is represented by the light blue hue. Both are multiplied, and when the final product values are combined together, the result (highlighted in light orange) represents an input value into the output feature map. Instead of padding the input image as in the last example, the kernel (represented by the chosen step-size) is given a stride of one in all vertical and horizontal places. Additionally, you can utilize a different stride value. Another consequence of raising the feature map's size is that it becomes smaller. Figure 1 will explain the mathematical operation of Convolutional layer as shown in below:

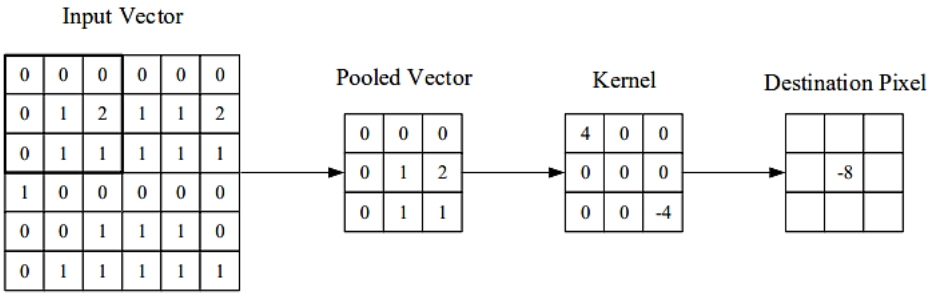

Fig. 1- Show the operation of the Convolutional layer[32] .

The convolutional layer uses a filter or kernel to apply a mathematical operation called convolution to an input matrix, extracting local features like edges, and generating a feature map. This process enables CNNs to be highly effective in computer vision tasks like image classification, object detection, and image segmentation.

Pooling Layer: Subsampling is the main function of the feature map pooling layer. Convolutional methods are used in the creation of these maps. Put another way, this method creates smaller feature maps from bigger ones. Furthermore, much of the dominating information (or attributes) is kept throughout the pooling process. The pooling operation is preceded by the assignment of the stride and kernel size. Various pooling layers can employ a variety of pooling algorithm types. These include min pooling, max pooling, average pooling, global max pooling, and global average pooling (GAP). The maximum, minimum, and GAP techniques are the most often utilized pooling algorithms. Some of these types are shown in Figure 2.

Fig.2- Show the Max pooling [33]

Fully Connected layer: The fully connected (FC) layer is usually located near the end of CNN topologies. The FC layer technique connects each cell in this layer to every other cell in the layer above. It functions as a classifier for CNN. As a feed-forward ANN, it uses the same fundamental method as a traditional multi-layer perceptron neural network The previous pooling or convolutional layer provides input to the FC layer. The form of this input is represented by a vector created from the flattened feature maps. Figure 3 shows that the FC layer's output is a representation of the final CNN output.

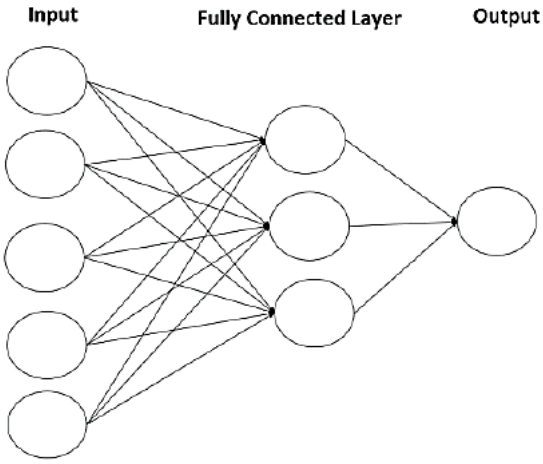

Fig. 3- Show the Fully Connected layer[34] .

4. Dataset

As seen in Fig. 5, it is also referred to as MalIMG and consists of a collection of 19506 known malware image files that span 31 distinct families. The size of the dataset is around 1.11 GB. The malimg dataset directory, which contains 31 subdirectories containing malware families transformed into PNG pictures, is located after downloading the malimg_dataset.zip file. A 32-character hash value is used to uniquely identify each malicious picture file. With an average size of 0.12 MB per PNG file, the pictures vary in size from 64 to 1024 pixels width by 208 to 5334 pixels height [35].

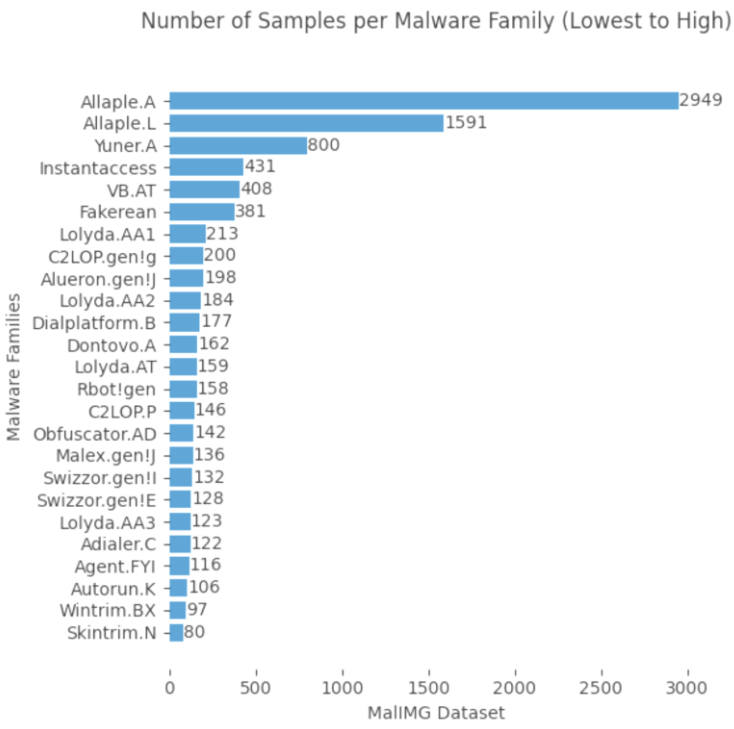

Fig. 4- Malimg Dataset Family[35]**.**

To sum up, the Malimg Dataset offers insightful information about the varied and changing world of malware. In order to train machine learning models that can recognize and categorize these threats, each of the 25 classes as shown in figure 4 represents a distinct collection of traits, behaviors, and attack vectors. Research on malware detection, classification, and prevention methods must advance if these classes are to be understood in an academic setting.

4.1. Data Augmentation

When the available dataset is limited or unbalanced, data augmentation serves as a strategy to enhance the diversity of data for training machine learning models. This process involves modifying existing data in ways that preserve the underlying patterns to create new data points. Data augmentation can be particularly beneficial in tasks such as speech recognition, image classification, natural language processing, and time series analysis. [36]. A key strategy for increasing the amount and diversity of training datasets is data augmentation, particularly in situations when there is an imbalance in the classes or a lack of data. Data augmentation enhances model generalization, robustness, and performance in a variety of fields, including computer vision, natural language processing, speech recognition, and time series analysis, by implementing diverse changes to the original data. The sorts of augmentations used must be carefully considered, though, as ill-chosen methods may cause the model's performance to deteriorate. Data augmentation will continue to be a crucial tool in the development of dependable and scalable models as machine learning and artificial intelligence advance [37].

4.2. Evaluation Metrics

Finding out how well-trained classifiers or learning algorithms perform on various data sets is the aim of assessment in deep learning. Most of the measures now in use concentrate on a classifier's ability to recognize classes. Metrics for evaluating classification performance must direct classifier development. There are significant problems with even the most widely used techniques, including figuring out the accuracy or error rate on a test set. Consequently, there is some association between modifications to classification algorithms and criteria

optimization. A lot of work has gone into creating ever-more-complex algorithms to deal with the categorization issue. At least as important as the algorithm is the assessment metrics phase, which comes first in the learning process [38]. Classifier performance may be measured in two ways: graphical and numerical. In contrast to numerical assessments, which provide a classifier's performance as a single number, graphical approaches display performance on a two- or three-dimensional plot, which facilitates human verification. While cost curves are examples of graphical methodologies, numerical performance evaluations include accuracy, precision, recall, and F1-Score.

The percentage of correctly classified information compared to all records is known as "accuracy" [39].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

In this case, True Positive (TP) represents the number of malware samples that were correctly categorized into the families they belong to, True Negative (TN) represents the number of non-malware samples that have been correctly identified as non-malware, False Positive (FP) represents the number of non-malware samples that were mistakenly identified as malware, and False Negative (FN) represents the number of malware samples that were incorrectly classified as their respective families [33]. The precision is determined by dividing the total number of samples categorized in a family by the percentage of correctly classified samples. When it comes to classifying malware [40].

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall: The percentage of correctly identified samples in a given family relative to the total number of true samples in that family is sometimes referred to as sensitivity[31] .

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

5. Results and Discussion

Unlike previous research, we suggested a 2D-CNN that makes use of a data augmentation technique to optimize the Malimg dataset's potential. In the past, classifiers have been trained using a variety of dataset-related characteristics. In order to facilitate autonomous feature extraction straight from the pictures, we suggested training the 2D-CNN. Our solution performs better than these earlier methods by using data augmentation techniques and training for 10 epochs . The Computational cost is depend on the Colab. This approach successfully prevented overfitting while enabling us to utilize the dataset to its fullest potential, as seen in Table 1.

Table 2 - Comparison Between different studies

Method	Accuracy of method
DenseNet-based model [18]	**98.23 %**
VGG16 [19]	**98 %**
VBDN [21]	**94.22 %**
Proposed Model	**98.86%**

As illustrated in the table 1, the proposed model outperforms the other techniques, demonstrating its versatility and resilience in Malimg dataset classification. Our architecture is specifically optimized for this application, as seen by the small improvement over other techniques. In comparison, DenseNet-based model [18], VGG16 [19], and VBDN [21] show a significant performance gap with accuracies of 98.23 %, 97 %, and 94.22 % respectively. It can be inferred from this that Malimg-based image classification is rather complex and for traditional ML models, it is impossible to implement properly. As previously mentioned, the accuracy of the performed tasks, for instance, in malware detection, security threat analysis, image recognition, can be trusted as reliable due to the proposed model's significant increase in classification performance to 98.86%. The Malimg dataset was identified with high

accuracy of 98.86%, which illustrates how well the proposed model performs and its ability to tackle a difficult problem. This clearly shows how successful our approach is when compared to prior

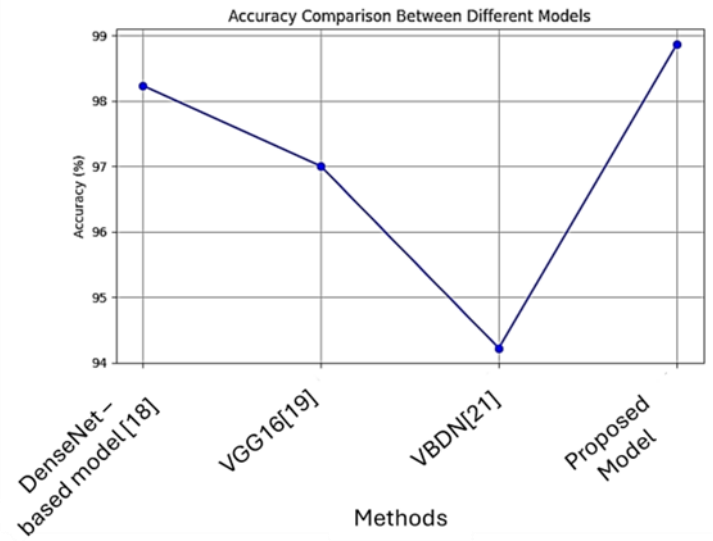

Fig. 5- Show the Comparison Between different studies

Figure 5 presents the classification accuracy comparisons of four approaches: the Proposed Model, the VGG16 [19], the DenseNet-based model[18], and VBDN [21]. The techniques can be seen along the x-axis, while their percentages for accuracy are taken along the y-axis. This upward trajectory that the suggested model takes on this curve illustrates an extraordinary boost in accuracy compared to the state-of-the-art approaches. The red flag has been drawn to higher performance in the suggested model, underlining how well it handled the intricacies of Malimg-based classification tasks.

This graphical representation, therefore, depicts that the proposed architecture is indeed robust and reliable to serve the purpose of enhancement in classification performance for optimization in malware detection and picture identification. The results clearly indicate that the suggested model raises the bar for accuracy in this field.

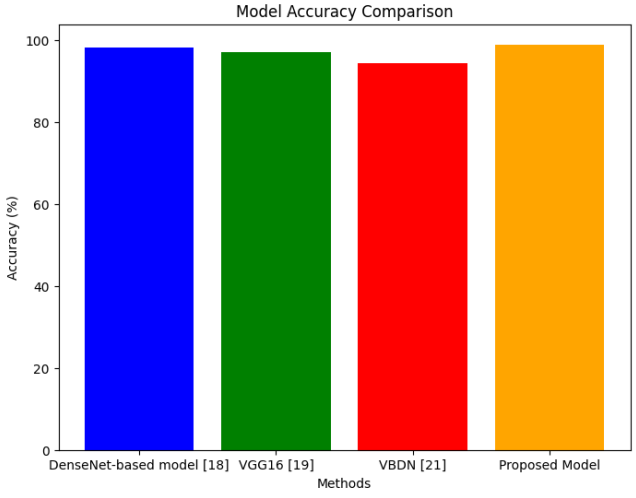

Fig. 6 - Comparison as Bar Blot

The accuracy of four distinct approaches used on the Malimg dataset—the DenseNet-based model [18], VGG16[19] , VBDN[21], and the Proposed Model—is graphically compared in Figure 6 as a bar plot.

With the suggested model achieving 98.86% accuracy, the bar plot successfully illustrates the steady increase in classification accuracy. This outperforms VGG16 by 1.66%, VBDN by 1.86%, and the DenseNet-based model by 0.76%.

The relative heights of the bars and the use of different colors highlight the resilience and optimization of the suggested model for Malimg dataset categorization while illuminating its comparative effectiveness. This figure provides compelling evidence that the model outperforms current approaches in handling the complexity of malware detection jobs.

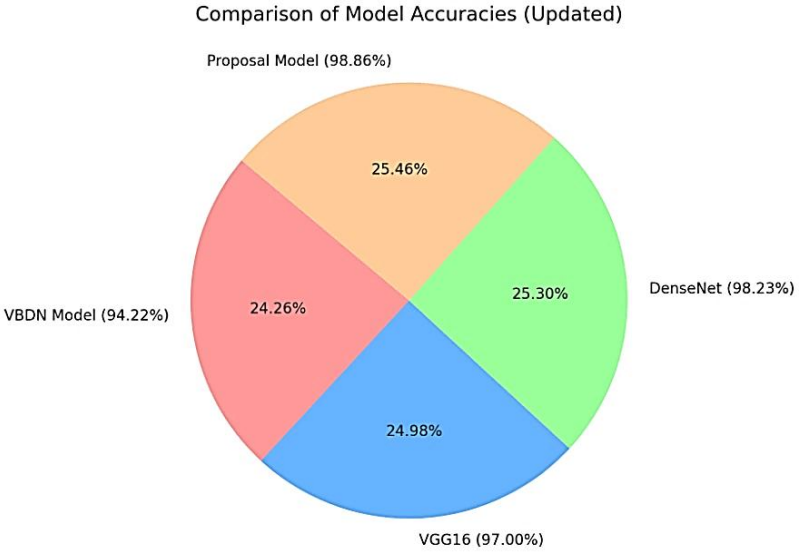

Fig.7- Pie chart

The similarity on the pie chart as shown figure 7indicates that the proposed model outperforms other methods on the Malimg dataset and provides a visual insight into the relative performances of the four strategies. The results emphasize the importance of domain-dependent optimizations in machine learning tasks, particularly in challenging domains such as malware detection, where generic methods, such as DenseNet[10], VGG16 [8], and VBDN model, provide reduced accuracy. This higher performance obtained using the proposed model indicates the strength of specialized architectures in capturing the subtleties of malware image classification, maximizing this model's potential for real-world applications in areas such as malware detection, threat analysis, and cybersecurity.

This research developed a CNN structure with a straightforward but powerful design. It included one convolutional layer, one max-pooling layer, and three connected layers each having 256 neurons. The convolutional layer extracts features helping the model understand spatial relationships in the input data. The max-pooling layer cuts down the dimensions, which makes the model stronger and speeds up calculations.

In addition to the network architecture, data augmentation techniques were applied to improve the generalization capability of the model. Specifically, zooming and flipping augmentations were employed, which artificially expanded the dataset by generating variations of the original images. This process effectively prevented the model from overfitting, as it introduced more diversity in the training samples, allowing the model to learn more generalized features. As a result, the accuracy of the model increased significantly, demonstrating the importance of augmentation in improving the performance of deep learning models, especially when dealing with limited training data.

Overall, the combination of a simple yet effective CNN architecture and the strategic application of data augmentation contributed to the model's high accuracy and robust performance.

This research shows that although conventional models work well, creative and customized methods—like the suggested model—are essential for addressing complex classification tasks in specialized datasets such as Malimg.

6. Limitations of Malware

Despite the usefulness of the Malimg dataset for image-based malware detection, there are certain boundaries that need to be examined, especially for business or research purposes. One major drawback is the limited number of malware types that the dataset covers. The dataset does not have all the forms that are present in actual cyber threats and, instead, focuses on a fragment of the well known malware. This dataset, as all others, evolves as malware morphs, and it could not capture the latest varieties or methods of attacks. In addition, some classes of the dataset may dominate, which means that an unproportionate amount of data for certain families of malware is included, while there are families of malware with lesser samples. Class imbalance, in turn, can lead to biased models. The imbalance may lead to the models that are more successful in the majority class but totally fail in the minority class. Preprocessing bias is another challenge that can happen when transforming malware binaries to images as it may the loss of crucial information. The dataset also does not precontaminate the samples with noise that is irrelevant, for which reason no superfluous software or anything else that a model would encounter in a real-life situation is limited.

Due to the fact that models that have been trained on the dataset may not operate well in more complex, real-life scenarios, they may produce results that are overly optimistic. Additionally, examples designed to deceive machine learning models are missing from the collection. The models of the Malimg dataset may be of limited use against adversarial attacks asmore sophisticated evasion techniques are employed by malware writers. Its over-reliance on image-based representational is an additional source of concern, as it is possible that it is not the most optimal way to analyze malware under all circumstances. For some types of malware, other models of feature extraction such as static or dynamic binary analysis could be more effective, even though deep learning models using image inputs are highly efficient. Moreover, another possible consequence of the enlarged size of the dataset is that it may not have enough diverse samples for training deep learning models which can deal with new or more complex threats. Finally, the lack of behavioral analysis annotations results in the dataset containing only static views of malwaresh and not the context of the virus action or its interaction with the system As a result, incorporating that information into models that make use of dynamic or behavioral analysis is much harder. Studying malware detection using Malimg as a benchmark is effective, but to continue remaining successful against emerging malware threats, shortcomings like old samples, preprocessing bias, or class imbalance, and simplistic reality need to be addressed.

7. Conclusion

A novel malware detection and family classification is proposed, adopted on a 2D-Convolutional Neural Network (CNN) optimized via data augmentation techniques as an effective methodology for malware detection in the context of Internet of Things (IoT) environments. Results from the Malimg dataset indicate that the proposed method outperforms previous methods that rely on a number of characteristics related to the dataset, thus demonstrating the method proposed can extract all features automatically, directly from images. By applying data augmentation and running for 10 epochs, attaining a 98.86% accuracy rate. A critical step in machine learning model optimization is hyperparameter tuning, which includes variables like learning rate, batch size, and optimizer selection. During training, the learning rate (LR) regulates the size of weight updates; a high LR can lead to instability and divergence, while a low LR causes delayed convergence. Grid search, random search, and adaptive scheduling strategies including step decay, cosine annealing, and exponential decay are all useful tuning approaches, this is depend on the type of the optimizer. The amount of data processed prior to updating the model weights is determined by the batch size, which affects generalization and training effectiveness. Larger batch sizes (e.g., 128–512) speed up training but may result in overfitting, whereas smaller batch sizes (e.g., 16–32) slow down training but frequently improve generalization owing to added noise, this value depend on the size of dataset. The optimizer selection, which dictates how model parameters are updated to minimize loss, is another crucial hyperparameter. When paired with momentum, SGD (Stochastic Gradient Descent) works well, but it needs to be carefully adjusted. In contrast, Adam dynamically adjusts learning rates, which makes it a popular option. Recurrent networks are especially well-suited for RMSprop, and AdamW outperforms Adam by adding weight decay to boost generalization. By fixing certain hyperparameters during training, computational cost is decreased, optimization complexity is decreased, and repeatability is guaranteed. In order to balance performance and resource efficiency and guarantee a well-trained model, effective tuning techniques concentrate on optimizing important hyperparameters, in general Adam is best choice for many researchers. This performance far outperforms other well-known techniques, such as Transfer Learning, Depth wise Efficient Attention Module (DEAM), and the DenseNet-based model. The results of this study highlight how crucial domain-specific optimizations are to tackling the difficulties associated with virus detection. The potential of CNNs as an effective cybersecurity tool is demonstrated by the suggested model's

capacity to manage the varied and dynamic nature of malware assaults. Moreover, comparison pie charts, bar charts and line graphs provided shows improved performance of our model in green colour. As the Internet grows by the day and the Internet of Things (IoT) takes off, the challenge of securing networked systems and discovering sophisticated malware attacks is only getting bigger. Despite being strong models, VGG16, DenseNet, and VBDN frequently need for a lot of data and a lot of processing power. For situations involving little data, computational limitations, and real-time applications, CNN2D + Data Augmentation provides a portable, flexible, and effective substitute. For image-based applications, CNN2D + Data Augmentation is frequently chosen over CNN + LSTM because of variations in generalization ability, computational efficiency, and architectural design. While LSTMs are meant for temporal sequence processing, CNN2D is especially made for spatial feature extraction, which makes it ideal for static picture categorization. Longer training periods and higher computing costs result from the needless complexity added by LSTMs to image classification jobs. Furthermore, because LSTMs are sequential, CNN + LSTM models need more memory, which reduces their efficiency compared to CNN2D. CNN + LSTM models need careful convolutional and recurrent layer tuning, which raises the possibility of instability, whereas CNN2D models are simpler to train in terms of optimization. Additionally, by adding dataset heterogeneity, data augmentation techniques like rotation, flipping, and scaling greatly enhance CNN2D models' resilience and generalization. Because LSTMs do not naturally use these augmentations, they perform worse on tasks involving static images. Because CNN2D does not analyze sequences sequentially as LSTMs do, it also provides quicker inference speeds. Furthermore, CNN2D is better suited for transfer learning with pre-trained models like ResNet and EfficientNet and is more scalable to huge datasets. Additionally, it is very well-suited for GPU parallel processing, while LSTMs have sequential dependencies that restrict computational parallelism. This work improves the strength and recognition capability of malware classification systems and assists ongoing efforts to secure cyberspace. Even though these results are promising, additional research is required to determine the proposed framework's scalability on alternative datasets and its capability for real-time applications for the identification of new risks. Future work will focus on adding this model to be more flexible in various conditions to efficiently improve detection accuracy by testing and integrating more complex feature extraction approaches or embedding hybrid models.This study essentially establishes the basis for more sophisticated malware detection systems by providing insightful information on how deep learning and domain-specific approaches may be used to address the escalating cybersecurity issues.

Reference

[1] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3041951.

[2] "On certain integrals of Lipschitz-Hankel type involving products of bessel functions," *Philos. Trans. R. Soc. London. Ser. A, Math. Phys. Sci.*, vol. 247, no. 935, 1955, doi: 10.1098/rsta.1955.0005.

[3] AV-TEST Institute, "AV-TEST Malware statistic," *Av-test*, 2021.

[4] A. M. Abbass and R. S. Fyath, "Performance investigation of geometric constellation shaping-based coherent WDM optical fiber communication system supported by deep-learning autoencoder," *Results Opt.*, vol. 15, no. August 2023, p. 100629, 2024, doi: 10.1016/j.rio.2024.100629.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning. Nature," *Nature*, vol. 521, no. 7553, 2015.

[6] "Statement of Retraction: A Survey on Malware Detection and Classification (Journal of Applied Security Research, (2021), 16, 3, (390-420), 10.1080/19361610.2020.1796162)," *Journal of Applied Security Research*, vol. 18, no. 3. 2023. doi: 10.1080/19361610.2022.2039530.

[7] M. I. P. Salas, P. L. de Geus, and M. F. Botacin, "Enhancing Malware Family Classification in the Microsoft Challenge Dataset via Transfer Learning," in *ACM International Conference Proceeding Series*, 2023. doi: 10.1145/3615366.3615374.

[8] A. S. Bozkir, A. O. Cankaya, and M. Aydos, "Utilization and comparision of convolutional neural networks in malware recognition," in *27th Signal Processing and Communications Applications Conference, SIU 2019*, 2019. doi: 10.1109/SIU.2019.8806511.

[9] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification," in *CODASPY 2016 - Proceedings of the 6th ACM Conference on Data and Application Security and Privacy*, 2016. doi: 10.1145/2857705.2857713.

[10] S. N. Swamy and S. R. Kota, "An empirical study on system level aspects of Internet of Things (IoT)," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3029847.

[11] M. Brahma *et al.*, "Learning impact of recent ICT advances based on virtual reality IoT sensors in a metaverse environment," *Meas. Sensors*, vol. 27, 2023, doi: 10.1016/j.measen.2023.100754.

[12] S. Kirmani, A. Mazid, I. A. Khan, and M. Abid, "A Survey on IoT-Enabled Smart Grids: Technologies, Architectures, Applications, and Challenges," *Sustainability (Switzerland)*, vol. 15, no. 1. 2023. doi: 10.3390/su15010717.

[13] D. Peraković, M. Periša, and P. Zorić, "Challenges and issues of ICT in industry 4.0," in *Lecture Notes in Mechanical Engineering*, 2020. doi: 10.1007/978-3-030-22365-6_26.

[14] N. Mangala, K. R. Venugopal, and B. Eswara Reddy, "Short Paper : Current Challenges in IoT Cloud Smart Applications," in *Proceedings - 2021 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2021*, 2021. doi: 10.1109/CCEM53267.2021.00016.

[15] D. Y. Kulkarni, G. Lu, F. Wang, and L. Di Mare, "Virtual gas turbines part I: A top-down geometry modelling environment for turbomachinery application," in *Proceedings of the ASME Turbo Expo*, 2021, vol. 2C-2021. doi: 10.1115/GT2021-59719.

[16] J. Al Faysal *et al.*, "XGB-RF: A Hybrid Machine Learning Approach for IoT Intrusion Detection," *Telecom*, vol. 3, no. 1, 2022, doi: 10.3390/telecom3010003.

[17] V. Gugueoth, S. Safavat, S. Shetty, and D. Rawat, "A review of IoT security and privacy using decentralized blockchain techniques," *Computer Science Review*, vol. 50. 2023. doi: 10.1016/j.cosrev.2023.100585.

[18] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for Malware detection," *Entropy*, vol. 23, no. 3, 2021, doi: 10.3390/e23030344.

[19] N. Pachhala, S. Jothilakshmi, and B. P. Battula, "Enhanced Malware Family Classification via Image-Based Analysis Utilizing a Balance-Augmented VGG16 Model," *Trait. du Signal*, vol. 40, no. 5, 2023, doi: 10.18280/ts.400534.

[20] H. Naeem, "Detection of Malicious Activities in Internet of Things Environment Based on Binary Visualization and Machine Intelligence," *Wirel. Pers. Commun.*, vol. 108, no. 4, 2019, doi: 10.1007/s11277-019-06540-6.

[21] F. M. J. Mehedi Shamrat *et al.*, "An advanced deep neural network for fundus image analysis and enhancing diabetic retinopathy detection," *Healthc. Anal.*, vol. 5, 2024, doi: 10.1016/j.health.2024.100303.

[22] M. ElKashlan, M. S. Elsayed, A. D. Jurcut, and M. Azer, "A Machine Learning-Based Intrusion Detection System for IoT Electric Vehicle Charging Stations (EVCSs)," *Electron.*, vol. 12, no. 4, 2023, doi: 10.3390/electronics12041044.

[23] N. Bhodia, P. Prajapati, F. Di Troia, and M. Stamp, "Transfer Learning for Image-based Malware Classification," in *International Conference on Information Systems Security and Privacy*, 2019. doi: 10.5220/0007701407190726.

[24] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020*, 2020. doi: 10.1109/ICAIIC48513.2020.9064976.

[25] A. Krishnan and S. T. Mithra, "A Modified 1D-CNN Based Network Intrusion Detection System," *Int. J. Res. Eng. Sci. Manag.*, vol. 4, no. 6, 2021.

[26] L. Zhao *et al.*, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, 2020, doi: 10.1109/TITS.2019.2935152.

[27] M. Zhan, J. Gan, G. Lu, and Y. Wan, "Graph convolutional networks of reconstructed graph structure with constrained Laplacian rank," *Multimed. Tools Appl.*, vol. 81, no. 24, 2022, doi: 10.1007/s11042-020-09984-2.

[28] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Networks*, vol. 8, no. 1, 1997, doi: 10.1109/72.554195.

[29] Stepanov S, Spiridonov D, Mai T (2023) Prediction of numerical homogenization using deep learning for the Richards equation. J Comput Appl Math 424:114980

[30] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, 2020, doi: 10.1007/s10462-020-09825-6.

[31] Ji X, Yan Q, Huang D, Wu B, Xu X, Zhang A, Liao G, Zhou J, Wu M (2021) Filtered selective search and evenly distributed convolutional neural networks for casting defects recognition. J Mater Process Technol 292:117064.

[32] A. Saxena, "An Introduction to Convolutional Neural Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 943–947, 2022, doi: 10.22214/ijraset.2022.47789.

[33] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, vol. 2018-Janua, no. November, pp. 588–592, 2017, doi: 10.1109/ICCSP.2017.8286426.

[34] T. A. Kalaycı and U. Asan, "Improving Classification Performance of Fully Connected Layers by Fuzzy Clustering in Transformed Feature Space," *Symmetry (Basel).*, vol. 14, no. 4, 2022, doi: 10.3390/sym14040658.

[35] 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.

[36] A. Roshanzamir, H. Aghajan, and M. Soleymani Baghshah, "Transformer-based deep neural network language models for Alzheimer's disease risk assessment from targeted speech," *BMC Med. Inform. Decis. Mak.*, vol. 21, no. 1, 2021, doi: 10.1186/s12911-021-01456-3.

[37] R. Hu, G. Ruan, S. Xiang, M. Huang, Q. Liang, and J. Li, "Automated Diagnosis of COVID-19 Using Deep Learning and Data Augmentation on Chest CT," *medRxiv*, 2020.

[38] A. A. Taha and A. Hanbury, "Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool," *BMC Med. Imaging*, vol. 15, no. 1, 2015, doi: 10.1186/s12880-015-0068-x.

[39] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, 2020, doi: 10.1186/s12864-019-6413-7.

[40] H. M and S. M.N, "A Review on Evaluation Metrics for Data Classification Evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, 2015, doi: 10.5121/ijdkp.2015.5201.