Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
June 2024 5(6), 1823–1830

MALWARE IMAGE PREDICTION AND CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK

Dr. P. Ramya[1], Manikandan S[2], Naveen G R[3], Madanbabu S[4], Madhankumar N[5]

[1]M.Tech., Ph.D., Associate Professor, Department of Computer Science and Engineering, Mahendra Engineering College, Namakkal
[2, 3, 4, 5] UG Students, Department of Computer Science and Engineering, Mahendra Engineering College, Namakkal

ABSTRACT

Without users' permission, malware software can infect computers or other devices. Through these loopholes, criminals commit a range of illegal and criminal offences that violate the legitimate rights and interests of the nation. Traditional malware categorization techniques fall into two categories: static analysis and dynamic analysis. It is usually not necessary to execute malware binary samples in order to perform static analysis techniques, and disassembly makes it simple to recover important data such as text lists, routines, and hash values. The static analysis methods offer a high accuracy rate and a simple operation with a low consumption time. Static analysis tools, however, are limited to analyzing malware binary samples at the surface level, where they are readily influenced by deformation and other means of confusion. Furthermore, it is challenging to identify and categorize unknown malware. Methods of dynamic analysis are not impacted by obfuscation and can operate in a virtual environment. It has the ability to recognize newly discovered malware samples and track the dynamic alterations of malware binary samples over time. Nevertheless, it's an extremely intricate and time-consuming process.

Malware has become one of the largest security threats in recent years due to its rapid growth. But feature engineering makes it difficult to handle large amounts of malware and readily limits the use of standard machine learning methods for malware categorization. However, dynamic analysis methodologies are not appropriate for efficiently categorizing large amounts of malware due to their complexity and high cost. In light of this, we propose a novel static malware detection method based on the convolutional neural network (CNN) employed in this work. Unlike existing methods, we use the data enhancement method to fix the unbalanced datasets, turn every viral byte into a colour image, and provide a better design.

Keywords: Malware image classification, Machine learning, Deep learning, Features extraction, Classification

DOI
10.29121/shodhkosh.v5.i6.2024.2602

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

How to cite this article (APA): Ramya, P., Manikandan, S., Naveen, G. R., Madanbabu, S., and Madhankumar, N. (2024). Malware Image Prediction and Classification Using Convolutional Neural Network. *ShodhKosh: Journal of Visual and Performing Arts, 5*(6), 1823–1830. doi: 10.29121/shodhkosh.v5.i6.2024.2602

1. INTRODUCTION

Malware, an acronym for malicious software, is a ubiquitous hazard in the digital realm that canjeopardise system security, privacy, and data integrity. As cybercriminal tactics evolve, the detection and mitigation of malware become increasingly critical. While traditional approaches predominantly focus on analysing code-based signatures or behavioural patterns, the rise of image-based malware poses new challenges. An emerging area of cybersecurity is malware picture classification, which uses machine learning methods to recognise and classify malware based on visual representations. This method enhances conventional detection techniques by closely examining photos linked to illicit activity, such as ransom ware notes, phishing website screenshots, or malware payloads. Malware attacks can target Internet of Things devices and smart appliances in addition to traditional PCs with Internet connectivity. Therefore, to defend millions of IoT users from harmful assaults, sophisticated cybersecurity techniques are required. Malware detection techniques have evolved over time. These vary from laborious manual labeling to intricate hybrid systems. Anti-malware software analyzes malware using data mining, association rule mining, information retrieval and extraction, and other approaches. By using these techniques, the amount of new malware being produced has increased dramatically. Static and dynamic analysis is the most often used methods for classifying malware. While dynamic analysis involves seeing malware in action, static analysis gathers data from malware programs without ever running them. Dynamic analysis is thought to be more reliable and efficient in the long run, despite its many drawbacks. For example, it cannot be instantaneously deployed at the endpoint because it takes too long to produce results—the malware requires time to assess itself before it can achieve its objective. Regretfully, the majority of these methods either heavily depend on feature engineering for their operation or demand specialized domain expertise for feature development. This is a concern since malware is developing considerably more quickly than this method can keep up with it. However, most popular anti-malware products combined the previously described techniques with a signature-based strategy that required creating a local database and keeping the infection's signature patterns.These sets of strategies, which were once useful in blocking various malware attacks, have now shown to be utterly inadequate in the face of newly created automated malware production techniques since they cannot keep up with the speed at which new malware is being created. Figure 1 depicts the fundamental malware.

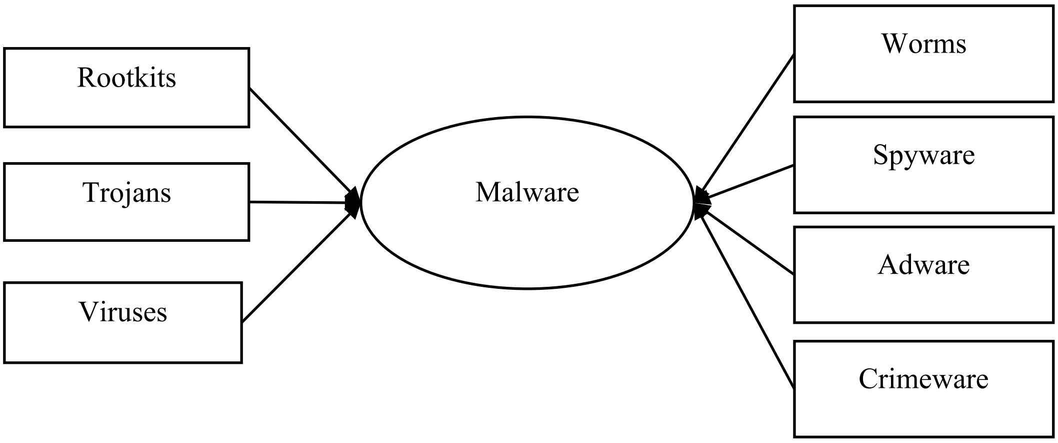

Fig 1: Malware image types

2. RELATED WORK

Mohamad mulham belal, et.al,[1] Applied picture visualization-based malware detection is growing in popularity because it can identify harmful activity on a computer system with greater accuracy. It is a helpful method for locating intricate and sophisticated malware that avoids detection through traditional channels. Rather than concentrating on code analysis or behavioral patterns, this technique evaluates potentially harmful actions by displaying malware samples. The RGB or grayscale images of the executable files or network traffic are further analyzed by the classification model using visual criteria to determine whether or not they are malware. Compared to traditional security techniques, malware detection based on image visualisation has several advantages. It is initially more accurate in identifying hazardous behaviour. In addition, it is quicker and simpler to utilize than conventional techniques. The technique makes use of a Self-Supervised ViT that has been trained on a large corpus of Android application (APK) files. SHERLOCK examines an APK file's representation to detect the presence of malware. In this paper, we suggested the Butterfly Vision Transformer (BViT) architecture as a vision transformer for malware image categorization.

The shortcomings of the ViT architecture and image malware-based research are addressed by the suggested architecture, BViT. The proposed malware classifier captures both the local and global spatial representations of malware pictures. For this, domain specialists are not required. Parallel processing of viral pictures is possible. It is also data-satisfied, scalable, and efficient in terms of time.

Peppes, Nikolaos, et al.,[2] creates and assesses a unique two-step procedure. Firstly, Deep Convolutional Generative Adversarial Networks (DCGANs) are used to build techniques for producing adversarial picture samples based on malware images that mimic the original ones. This promotes the creation of a deep convolutional neural network model that employs transfer learning to recognize these kinds of questionable images by making more (otherwise unusual) training and validation datasets available. This study presents a way to address malware detection issues (i.e., lowering the likelihood of malware evading systems) that consists of two primary parts: i) DCGANs are used to create malware-based suspicious photos; ii) Deep convolutional neural networks are used in combination with transfer learning approaches to detect suspect photographs. Next, utilizing FMGAN, a next-generation malware detector is produced that is incredibly adaptive to freshly created adversarial samples and malware concealing tactics. This can be done by using data that the FMGAN can reliably supply to create an ongoing training process for the detection model. Therefore, the study's additional value is found in the way it provides a detailed explanation of the system's design and development, which involves comparing multiple CNN architectures and training datasets to determine the best configuration. This entails using the FMGAN to create adversarial samples and putting effective malware detection into practice. It occurred to us to design a detector specifically trained on fashion products, given the sizeable market for e-commerce products.

Kim Jeongwoo, et al.,[3] provide a state-of-the-art, cross-modal attention-based approach for classifying malware families. Finding malware families in a non-disassembled file (a binary file) is the goal, and it is accomplished by making full use of the static properties of many modalities. Since we focus on non-disassembled data, the proposed methodology can manage malware without considering disassembly issues. We leverage malware pictures and malware structural entropy in our multi-modal method. The former is displayed using binaries with byte granularity, whereas the latter is computed from binaries with chunk granularity (512 bytes, 1024 bytes). Numerous malware analyses have shown how successful each strategy is. Furthermore, using both can counterbalance each other out: structural entropy, which is decided at the chunk level, is resistant to byte-level change, whereas malware pictures are susceptible to byte distortions caused by code obfuscation techniques. However, particular information about byte-level patterns is lost in structural entropy. Malware images, on the other hand, retain byte-level information within their pixels. We created a novel attention-based cross-modal convolutional neural network (CNN) with a combination of malware images and structural entropy in order to optimize the synergistic effect. Two different CNNs for images and structural entropy were used to create the corresponding intermediate representations.

Faiza Babar Khan and others. [4] focuses on feature extraction and the vanishing gradient problem in malware few-shot classification using a novel architecture called ResRelNet (RRN). The convNet's hereditary structure is strengthened by the residual connections of the RRN architecture, which are based on Residual Neural Networks and allow for a more fluid information flow between layers. This design strategy fixes the fading gradient problem in deep networks and improves the extraction of high-level and useful PE features. Moreover, the proposed model is trained by a series of discrete learning tasks, or "episodes." The model can identify frequent visual patterns and aspects pertinent to the tasks at hand because of the episodic nature of the training process. Our suggested methodology is unique in the field because it uses ResNet for transfer learning, episodic training, and getting beyond FSL restrictions. To identify the variables that improve accuracy, the tests are conducted in several stages using various graphs, and the outcomes are compared to the baseline approach. Since the concept's original introduction, FSL has been recognized as a useful method for identifying classes that are not visible during training with a small number of labeled instances. Finding new malware is a common challenge in FSL, as proficient virus developers make obtaining training samples with supervised data exceedingly challenging. Small code modifications can have an impact on malware hashes, and unique malware can make it impossible to use numerous real records for training. Consequently, it is crucial to draw meaningful conclusions from a small number of samples. By means of FSL, the model has the ability to gather comprehensive and intricate malware data through a sequence of comparable activities. It can adapt to classes it has never met in the future by compiling task-specific information from a small number of samples.

Geremias, Jhonatan, et al., [5] suggests a novel, dependable, two-stage hierarchical CNN model for Android malware classification based on images. Initially, Android malware is categorized using an image-based CNN in a hierarchical local classification context. As a result, on a parent node, examined programs are first categorized as benign or harmful samples. On a child node, a tailored CNN model locates the family of maliciously classified apps. To improve the classification reliability in a reject-option classification, the child node is only assigned the jobs from the parent CNN node that are most confidently classified. Such a concept has the insight that, in order to keep the system accurate, only malware samples with a high degree of confidence need to have their family identified. We're still early in our work on Android malware detection using CNNs for image classification. In this research, we present a novel image-based CNN technique for reliable hierarchical Android malware detection that may be used to identify app families that may pose a threat. Our proposed method rejects very few app samples, but can improve the detection rate of malware programs. Moreover, it can increase the average detection rate of the known malware families when compared to traditional approaches.

3. EXISTING SYSTEMS

For malware picture classification, the k-Nearest Neighbors (k-NN) algorithm is a straightforward yet effective method of classifying images that depict different types of infections. Initially, a dataset containing a diverse range of malware images labeled with the relevant malware category or kind is put together. Then, using methods like local binary patterns (LBP), histogram of oriented gradients (HOG), and deep learning-based feature extraction, these images are converted into numerical feature vectors. When malware photos are analyzed using the k-Nearest Neighbors (k-NN) algorithm, a methodical approach to classifying malware images based on their resemblance to others within a dataset is taken. Initially, a diverse collection of malware images is assembled, with each image labeled according to its respective malware type or family. These images undergo feature extraction, where relevant characteristics are numerically represented. To do this, techniques like the histogram of oriented gradients (HOG) and local binary patterns (LBP) are commonly employed, enabling the capture of distinctive visual traits. Normalisation is used after feature extraction to guarantee that every feature makes an equal contribution to the distance computation in the k-NN algorithm and to guard against biases. The performance of the model is then evaluated by training and testing subsets of the dataset. During training, the k-NN technique determines the distances between the features of each image in the training set and the unseen features of the photos in the testing set. Next, labels are assigned among the k nearest neighbours according to the majority class. By means of this repeated process, the model acquires the ability to distinguish between distinct kinds of malware according to their visual attributes. All things considered, using the k-NN algorithm for malware image analysis offers a reliable methodology for locating and classifying malware, supporting continuous efforts in cybersecurity to reduce emerging threats.

Additionally, learn how to maximize the margin—the distance between the hyperplane and the nearest data points from each class—while minimizing classification errors using the Support Vector Machine technique. To accomplish this optimization, various parameters are changed, including the regularization parameter, kernel parameters, and kernel function (polynomial, radial basis function, etc.). By mapping new malware images into the same feature space and calculating their position in relation to the decision border, the SVM model can effectively identify them once it has been trained. This classification enables the SVM to assign labels to the input images, identifying the type of malware they represent. SVMs offer a robust and flexible approach to malware image classification, capable of handling high-dimensional data and nonlinear decision boundaries with high accuracy and efficiency.

4. NEURAL NETWORK BASED MALWARE IMAGE ANALYSIS

Convolutional Neural Networks (CNNs) provide a cutting-edge approach to malware picture classification through deep learning techniques. CNNs do this by automatically extracting and evaluating complex visual features directly from raw pixel data. CNNs using this technology are made up of several layers that are intended to gradually acquire hierarchical feature representations from input images. Convolutional layers first employ filters to identify regional patterns and characteristics, obtaining crucial visual cues suggestive of various malware kinds.The feature maps are then down sampled by pooling layers, which reduces computational complexity while maintaining important information. The ultimate classification judgment is subsequently reached by fully linked layers after they have combined and interpreted these acquired features. Large datasets of annotated malware images are used to train the CNN. During this process, the network modifies its internal parameters using gradient descent and backpropagation, two iterative optimization approaches. This makes it possible for the CNN to effectively differentiate between different malware kinds and reduce classification errors. To improve the model's resilience and generalization, the training dataset is also exposed to data augmentation methods like rotation, scaling, and flipping. Through the use of pre-trained models that have been refined through extensive picture dataset training, transfer learning can further enhance CNN performance. The process for this job is described below.

Prior to processing: With this module, we may scale the image and remove noise by applying the median filtering technique. The filter's goal is to eliminate noise that has warped the picture. It is based on an analysis of statistics. Common filters are designed with a particular frequency response in mind. Filtering is a nonlinear technique in image processing that is commonly used to reduce "salt and pepper" noise. If reducing noise and maintaining edges is the goal, a median filter works better than convolution. The first step in image preprocessing with the median filtering algorithm is to import necessary libraries, including OpenCV and NumPy, to make image manipulation easier. After loading the target image, it can be converted to grayscale optionally to make further processing steps easier. The cv2.medianBlur () function is used to apply the median filtering approach, which is the main component of the preprocessing. This function requires the kernel size to be specified, as well as the neighbourhood that is utilised to compute the median values. An important factor influencing the trade-off between noise reduction and image smoothness is selecting an odd kernel size. and complete homework involving photo binarization. Image binarization is a phase in the malware image analysis preparation process that separates the foreground image from the background.

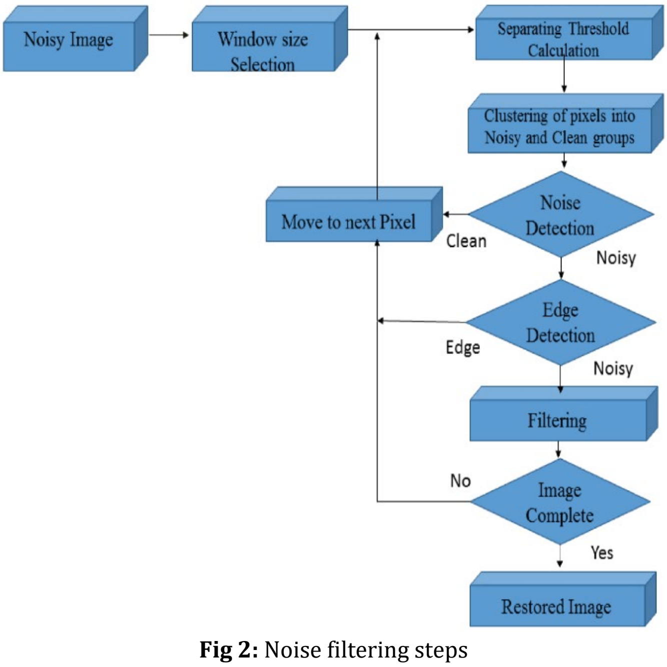

Fig 2: Noise filtering steps

ALGORITHM FOR MEDIAN FILTERING

INPUT: Size M x N Image X with filter size n

OUTPUT: Y image with X's dimensions

```
Set the initial values of the histogram H for the following cases:
I = 1 to M, j = 1 to N,
K = -n/2 to n/2.
Take Xi+k, j-n/2-1 out of H.
Add Xi+k, j+n/2 to H,
The end for Yi,j,
The median (H), and
Finish
```

MODEL CONSTRUCTION AND PREDICTION

Feed-forward networks with the ability to recognize topological features in an input image are called convolutional neural networks. A classifier then uses the gathered information to categorize the raw image. CNNs are unaffected by basic geometric alterations like as translation, scaling, rotation, and squeezing. Convolutional neural networks employ three architectural principles—shared weights, local receptive fields, and spatial or temporal sub-sampling—to attain different levels of shift, scale, and distortion invariance. Similar to a traditional neural network, back propagation is typically used to train the network. A convolutional layer uses the local receptive fields from the preceding layer as a source of features. A 5x5 convolution kernel network has 25 inputs per unit connected to the 5x5 area of the previous layer's local receptive field.

All units inside a feature map have the same weights; however each connection is assigned a trainable weight. Weight sharing is incorporated into each layer of CNN, allowing for a reduction in the overall number of parameters that can be trained. Basic visual features like edges can be extracted by neurons from local receptive fields. It is possible to extract the same visual property using neural structures that have the same weights shared by neurons at different places. A feature map is the result of this kind of neural network. This process is equivalent to convolutioning the input image using a tiny kernel. It is possible to use numerous feature maps to extract different visual features from an image. Subsampling the feature map to lower its resolution decreases the output's sensitivity to shifts and distortions. From each original eye data set, many features can be retrieved using our suggested CNN structure, and each feature has dimensions.

- Convolutional Layers: Sequential framework consists of 13 convolutional layers, which are used to extract features from input images. These layers are followed by max-pooling layers that down sample the feature maps to capture hierarchical information.
- Fully Connected Layers: After the convolutional layers, it has three fully connected layers and an output layer for classification. These fully connected layers ultimately determine the class of the input image.
- Receptive Fields: it uses relatively small 3x3 convolutional filters in its layers. This architecture results in very small receptive fields for each neuron, allowing it to capture fine-grained details in the images.
- Stacking Convolutional Layers: The customized architecture is characterized by the repeated stacking of convolutional and pooling layers, which allows it to learn features at different scales.
- ImageNet Pretraining: The ImageNet dataset, which has millions of tagged photos in thousands of categories, served as the pretrained dataset. The model gains a thorough comprehension of a variety of visual ideas from this pretraining.
- Transfer Learning: Its pretraining makes it an excellent choice for transfer learning. And can fine-tune the model on a specific task, like liver disease detection, by replacing the last few layers while keeping the pretrained layers' weights intact.
- Deep Network: It is relatively deep compared to its predecessors and is capable of learning intricate features and patterns from images. However, this depth also results in increased computational complexity.

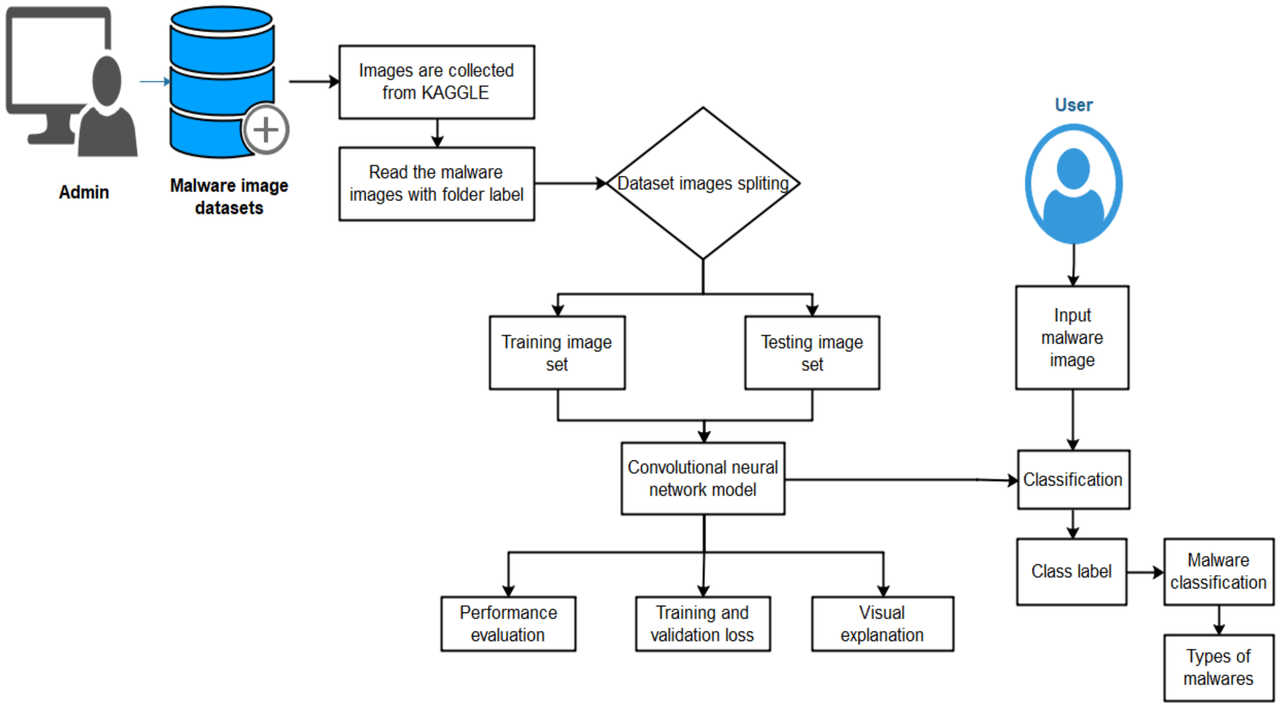

Fig 2: Proposed architecture

5. EXPERIMENTAL RESULTS

When assessing the effectiveness of a classification model, such as the ones used in malware image classification jobs, a confusion matrix is a helpful tool. It offers a concise synopsis of the discrepancies for each class between the actual ground truth labels and the model's predictions.

- True Positives (TP): In these instances, the model accurately predicts a malware image that falls under a particular category.
- False Positives (FP): These happen when a malware image is mistakenly predicted by the algorithm to belong in a given category when it actually does not.
- True Negatives (TN): In these instances, the algorithm accurately forecasts that an image that isn't malware falls outside of a specific category.
- False Negatives (FN): These happen when a non-malware image is mistakenly predicted by the algorithm to belong in a given category when it does not.

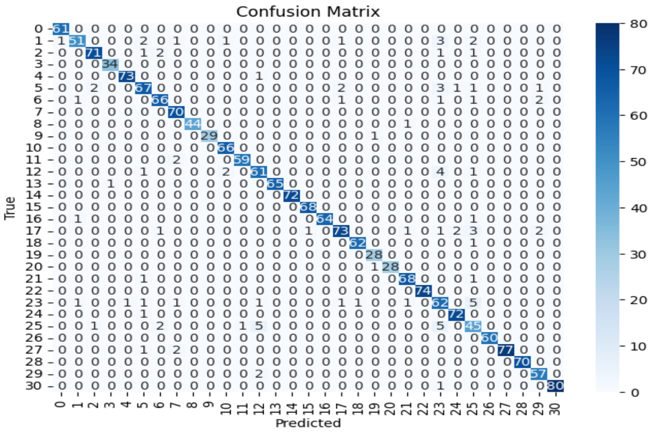

Fig 3: Confusion matrix

This figure shows that the proposed system provides improved true positive rate than the existing machine learning algorithms

6. CONCLUSION

In conclusion, malware image classification is a crucial task in cybersecurity, aiming to automatically detect and categorize malicious software based on visual representations. Numerous techniques have been used to address this issue, including traditional machine learning algorithms like Support Vector Machines (SVM) and k-Nearest Neighbours (k-NN) and deep learning strategies like Convolutional Neural Networks (CNNs). Every strategy has specific benefits and things to keep in mind. For example, because k-NN is simple to use and understand, it does well on baseline classification tasks. Because of their improved performance with high-dimensional data and nonlinear decision boundaries, support vector machines (SVMs) are an excellent option for difficult malware photo classification tasks. However, CNNs use deep learning to automatically extract hierarchical features from raw pixel data, which makes malware picture classification reliable and accurate. Cybersecurity experts can improve threat detection capabilities, reduce possible risks, and protect digital infrastructures from malicious threats by implementing these strategies. Moreover, the continuous advancement of machine learning and deep learning methodologies promises further improvements in malware image classification accuracy and efficiency.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Belal, Mohamad Mulham, and Divya Meena Sundaram. "Global-Local Attention-Based Butterfly Vision Transformer for Visualization-Based Malware Classification." IEEE Access (2023).

Peppes, Nikolaos, et al. "Malware Image Generation and Detection Method using DCGANs and Transfer Learning." IEEE Access (2023).

Kim, Jeongwoo, Joon-Young Paik, and Eun-Sun Cho. "Attention-Based Cross-Modal CNN Using Non-Disassembled Files for Malware Classification." IEEE Access 11 (2023): 22889-22903.

Khan, Faiza Babar, et al. "Detection of data scarce malware using one-shot learning with relation network." IEEE Access (2023).

Geremias, Jhonatan, et al. "Towards a Reliable Hierarchical Android Malware Detection Through Image-based CNN." 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC). IEEE, 2023.

Prajapati, Pratikkumar, and Mark Stamp. "An empirical analysis of image-based learning techniques for malware classification." Malware analysis using artificial intelligence and deep learning (2021): 411-435.

Singh, Jaiteg, et al. "Classification and analysis of android malware images using feature fusion technique." IEEE Access 9 (2021): 90102-90117.

Sharma, Gurumayum Akash, Khundrakpam Johnson Singh, and Maisnam Debabrata Singh. "A deep learning approach to image-based malware analysis." Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019. Singapore: Springer Singapore, 2020. 327-339.

O'Shaughnessy, Stephen, and Stephen Sheridan. "Image-based malware classification hybrid framework based on space-filling curves." Computers & Security 116 (2022): 102660.

Iadarola, Giacomo, et al. "Image-based Malware Family Detection: An Assessment between Feature Extraction and Classification Techniques." IoTBDS. 2020.