I.J. Information Engineering and Electronic Business, **2023, 2, 20-29**
Published Online on April 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2023.02.03

Malware Multi-Class Classification based on Malware Visualization using a Convolutional Neural Network Model

Balram Yadav*
Ph. D. Scholar, Computer Engineering, Institute of Engineering and Technology, DAVV, Indore-452017, India
E-mail: balram.dreamsworld@gmail.com
ORCID iD: https://orcid.org/0000-0001-5182-8324
*Corresponding Author

Sanjiv Tokekar
Professor, Director, and Head of Department, Electronics and Telecommunication Engineering, Institute of Engineering and Technology, DAVV, Indore-452017, India
E-mail: stokekar@ietdavv.edu.in
ORCID iD: https://orcid.org/0000-0002-0845-0300

Received: 26 August, 2022; Revised: 25 October, 2022; Accepted: 10 December, 2022; Published: 08 April, 2023

Abstract: Malware classification has already been a prominent concern for decades, and malware attacks have proliferated at an astounding rate, constituting a significant threat to cyberspace. Deep learning (DL) and malware image approaches are becoming more prevalent in the field of malware analysis, with spectacular results. This work focuses on the challenge of classifying malware variants that are represented as images. This study employs visualization and proposes a convolutional neural network (CNN) based DL model to effectively and accurately classify malware. The proposed model is trained and tested on a very challenging and heterogeneous dataset, and it achieves accuracy of 98.179%, precision of 97.39%, a F1-score of 97.70%, and a fast classification speed (3 seconds needed to test 934 unseen malware). This demonstrates the proposed model's incredibly quick, effective and accurate performance. The proposed model outperformed existing traditional DL models in terms of various performance measures and demonstrated its usefulness in classifying malware families through visualization. This study and experimental results reveal that small-scale malware images and a simple CNN architecture alone are capable of accurately classifying malware families with high classification accuracy.

Index Terms: Convolutional neural network, CNN, Deep learning, DL, DL models, Malware, Malware classification, Malware visualization.

1. Introduction

Malware is a term used to describe malicious software and is among the deadliest cyber threats today in the digital world [1]. Malicious software performs actions with the intent to harm computers, resources, information, and damage internet of things (IoT) systems [2]. Users are always concerned about computer system security. Malware classification is considered one of the most crucial cyber security challenges [3]. The daily rapid increase in malware samples or malicious attacks [4,5] and their nature pose a severe problem as well as a challenge for nations, economy, social society, and ordinary users [1].

Traditionally, malware is analyzed using static, dynamic, hybrid [6,7], data mining, and machine learning (ML) [8,9] based approaches. Static analysis removes the binary code from an executable file to provide patterns or features that can be used to identify malware. However, it has been shown that this methodology falls short when attempting to analyze various code obfuscation and packing techniques. These methods are time-consuming, require manual feature extraction, and have other disadvantages [8].

Recently, malware evolution statistical reports from antivirus companies (Refer Figure 1 and Figure 2) have shown an expedited increase in the amount of malware and its variant attacks. Reports from AVTEST Institute [4] and McAfee antivirus company [5] clearly describe that million of malware and its variants attacks are exposed and reported and the

report also reveals the exponential development of IoT malware, mobile malware, and a rapid increase in ransomware [1]. Intending to deal with this challenge, it is necessary to fight this issue; nowadays DL and its models have achieved a breakthrough to address malware classification. DL is a type of ML that is used to learn optimized features by analyzing enormous amounts of data, and automatically extracts [2] unknown or hidden representations of the data, as opposed to manually crafted extraction of representations of ML algorithms.

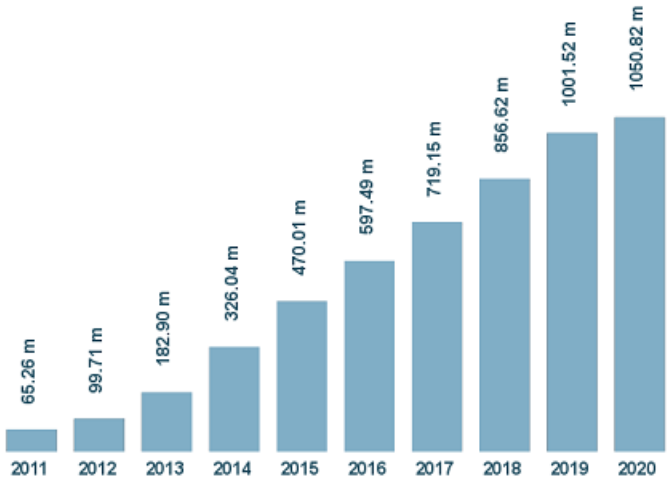

Fig. 1. Statistics on the proliferation of malware

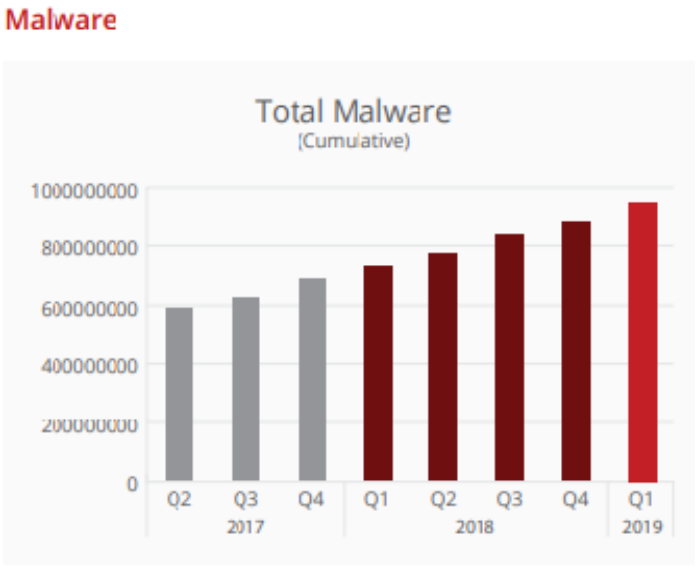

Fig. 2. Malware evolution cumulative statistics

DL architectures have shown great performance in many disciplines, like the medical field, computer vision, natural language processing, speech recognition, and achieved a breakthrough to discuss malware analysis [3] and [10]. ML requires the extraction of features from images for classification purposes, although DL is capable of accomplishing this task on its own. For image classification and vision-based processing, CNN is among the most widely used and current benchmark DL models [11-15] and it has shown a great performance to classify images with high accuracy [11], and [16-18]. Manual feature extraction is the most challenging aspect of earlier methods described in the literature. The compatibility of the extracted features affects how effective the process is. As a result, the suggested method successfully solves this problem by choosing the CNN to automatically extract deep features.

Malware classification is a vital step toward protection in the discipline of cyber defense. To classify malware into a family not only aids human experts and automated systems in analyzing samples from a certain malware family but also aids in the recommendation of certain preventative actions to limit malware attacks. The main objective of this work is to examine and propose a simple DL model for classifying malware using low-dimensional malware images.

This work's key contribution is the improvement of malware classification accuracy with the use of low-dimensional images and a fundamental CNN architecture. It presents a simple and cost-effective approach to classify malware and its variants automatically. It is a significant problem in image classification tasks since the complexity of

the classification model rapidly grows along with the training and testing time when the image size and deep layered architecture are implemented. As a result, this advancement is crucial for image classification, where malware classification is a crucial and severe issue. Towards this undertaking, this paper proposes the lightweight CNN model, which is a deep neural network (NN) for malware classification using visualization.

The paper is structured as follows. Section 2 reviewed the related work used in DL and visualization-based malware classification. The proposed methodology is described in detail in section 3. In section 4, the findings of the experiment are provided and finally, the research work is concluded in section 5.

2. Related Work

Recently, DL has been used to classify malware and is found in the literature to be more effective and accurate than traditional methods [19]. The literature review focuses on the research done on the methods and approaches using DL and malware visualization.

The research by [8] in 2011 was considered the first solution for malware visualization and the results validated the visualization's effectiveness. The authors transformed malware into images, extracted texture features, and then classified the images by calculating Euclidean distance using the K-nearest neighbour algorithm. Kosmidis and Kalloniatis [9] (2017) provided an automated NN-based framework for classification and used a variety of ML algorithms on it. They achieved an average accuracy of 0.0856 using the nearest centroid, which was the lowest result; however, they got a reasonable accuracy of 0.0916 using the random forest, which was the top result, with other ML methods bringing them intermediate accuracy. Mourtaji et al. [12] (2019) presented and designed a CNN model (they used the CNN architecture defined by Simonyan and Zisserman [20] (2014) to classify malware and achieved higher accuracy on grayscale images. For classification, they applied a pre-trained model; the visual geometry group (VGG-16), with the same initial weights and experimented with different train-test ratios, batch size, learning rate, etc.

Kabanga and Kim [16] (2017) implemented a three-layer CNN model for classifying malware images (with a size of 128x128). The authors used CNN because it is reliable, can be applied to an entire image immediately, and is best suited for automatic feature extraction. Kalash et al. [11] (2018), the authors implemented a deep CNN model; first, they translated the malware binaries into grayscale images. The authors have refined the hyperparameters of the high-performance, pre-trained VGG-16 model to classify very high-dimensional malware images, and achieved high accuracy on two known datasets. Hamad [2] (2019) addressed the problem of malware detection in IoT networks. The author developed a naive strategy for converting the malware binaries into very high dimensional colour images and then implemented a fast and 4-layer deep CNN model with numerous filters to target malware in IoT networks by using different image sizes to recognize it. Kumar et al. [13] (2018) used malware image similarity measurements to detect zero-day malware by implementing a 3-layer deep CNN model; they tested their proposed solution on three different types of data sets. They used malware binaries as grayscale images and concluded that the image processing method was useful to achieve better performance and accuracy. To turn malware binaries into images, they used a method by [19]. Lu and Li [21] (2019) implemented a generative adversarial network (GAN), one of the latest DL models, and addressed data imbalances. GAN is implemented using 18-layer deep CNN, although the results obtained are not promising, using GAN allowed them to generate more fake training examples and proved helpful in training small datasets.

Recently, Jain et al. [14] (2020) applied extreme learning machines (ELMs) and the 2-layer CNN model for the same task and compared CNN with ELM. They proved that ELMs took less time to train compared to CNN. ELMs proved to be faster than CNN and achieved higher accuracy in the 1D data processing. With ELMs, they should do very little experiment set up; they just tune some neurons and activation functions in the hidden layer. A more recent approach was proposed by Vasan et al. [18] (2020). The authors demonstrated a novel CNN architecture that focuses on an ensemble of CNNs. Because each CNN model differs by architecture, they have different visual semantics of images, and a collection of CNN models extracts the highly optimized features and greatly improves the malware classification compared to a single CNN architecture. Cui et al. [23] (2018) implemented a DL model to improve malware detection and used data augmentation approaches to address the class imbalance issue in their solution. First, they applied the grayscale translation of the malicious code; the images were then classified and identified by the 7-layer CNN model. The recommended method was successful and achieved higher accuracy and faster detection time. Agarap [6] (2017) implemented a Fusion of the DL model with the ML algorithm, using CNN to extract features from images and the support vector machine (SVM) as a classifier to classify the images. The classification was performed on different DL models such as multilayer perceptron (MLP), CNN, and gated recurrent unit (GRU). According to empirical evidence, the GRU outperformed the other DL models with a prediction accuracy of 84.92%.

In the literature, significant efforts have been made to automatically identify malware families using a variety of deep learning models and by combining DL models. There has been a lot of research in the literature on using deep learning to automatically classify families of malware. According to the study mentioned above, visualization-based techniques have established themselves as the latest trend, and deep learning automates this procedure. It has been noted that the malware visualization method aids in improving performance and classification accuracy. In summary, the adoption of DL methods for identifying and classifying malware by translating malware binaries into images is currently flourishing [1], and various DL models and architectures are being explored. There is a variety of DL

architectures available, new light weight DL models are being explored, and with different hyperparameter settings, more detailed investigations are needed to find solutions that are both efficient and precise in the malware classification domain.

3. Proposed Method

In this section, a CNN model is discussed and recommended for efficiently classifying malware images into different families. The proposed approach differs in that a DL model is applied to learn discrimination patterns from the images and classify them compared to [8]. Here the classification task consists of two modules. The first module deals with training, where the classifier is constructed from a collection of labelled data (malware images/family), followed by the second module, which deals with testing the classifier (where the classifier is based on a set of unlabeled images that were not seen during the training period).

3.1 Data set

Datasets are crucial in the training, testing, and validation of systems [1]. In this work, the proposed model's performance was tested and evaluated on a malware dataset [24]. It has 9339 grayscale malware images belonging to 25 unique malware families [1] and their variants (Table 1). For experimental purposes, the whole dataset was further divided into three subsets: the training set (80%), the validation set (10%), and the test set (10%). The distribution of malware samples in the data set is summarized in Table 1.

Table 1. Malimg malware dataset details

S. No.	Malware Class	Family Name	No. of Variants
1	Worm	Allaple.A	2949
2	Worm	Allaple.L	1591
3	Worm	Yuner.A	800
4	Dialer	Instant access	431
5	Worm	VB.AT	408
6	Rogue	Fakerean	381
7	PWS	Lolyda.AA1	213
8	Trojan	C2Lop.gen!G	200
9	Trojan	Alueron.gen!J	198
10	PWS	Lolyda.AA 2	184
11	Dialer	Dialplatform.B	177
12	Trojan Downloader	Dontovo.A	162
13	PWS	Lolyda.AT	159
14	Backdoor	Rbot!gen	158
15	Trojan	C2Lop.P	146
16	Trojan Downloader	Obfuscator.AD	142
17	Trojan	Malex.gen!J	136
18	Trojan Downloader	Swizzor.gen!I	132
19	Trojan Downloader	Swizzor.gen!E	128
20	PWS	Lolyda.AA 3	123
21	Dialer	Adialer.C	122
22	Backdoor	Agent.FYI	116
23	Worm:AutoIT	Autorun.K	106
24	Trojan Downloader	Wintrim.BX	97
25	Trojan	Skintrim.N	80
Total			9339

3.2 Dataset Investigation

The classifier's performance is degraded by an uneven dataset [17], [22] because minority class characteristics are not learned very well and efficiently, resulting in poor or absent classification [23]. In the dataset, Autorun.K, Wintrim.BX, Skintrim.N are minority classes, and classes Allaple.A, Allaple.C dominate the dataset. Table 1 shows the unbalanced distribution of the images in the respective families. Another observation from the data set is that the Autorun.K, Yuner.A malware families are structurally very similar and the sample file sizes are almost the same. The class imbalance ratio is defined [23] as

$$imbalance\ ratio = \frac{number\ of\ samples\ in\ the\ majority\ class}{number\ of\ samples\ in\ the\ minority\ class} \quad (1)$$

In the dataset, the majority class is Allaple.A (highest 2949 images) and the minority class is Skintrim.N (lowest 80 images). The imbalance ratio according to Eq. 1 of the dataset is:

$$\text{Imbalance ratio} = 2949/80 = 36.86 \text{ (approx. 36:1)}$$

3.3 Pre-processing

The recommended solution is largely influenced by malware visualization work based on the observation that images of different malware samples from the identical family would show similar visual similarity, while samples from different malware families would show differences. The pre-processing of the dataset used [24] requires two operations (resize and rename) on the raw images of the malware dataset before feeding them into the CNN model since the dataset used contains 9339 grayscale images of different sizes and requirements contain images of the same size [25], so the images are resized.

3.4 Proposed CNN architecture

The proposed approach uses CNN to learn a feature hierarchy from pixels to the layers of the classifier. Supervised learning is used to train the model. The CNN model building process is divided into two phases: the training phase and the classification phase. A two-layer CNN is used for the classification task; the model contains two convolutional (Conv) layers, followed by two Max-pooling layers, and then two fully connected (FC) layers. A typical CNN takes an input image, goes through several layers of processing—convolution, activation function, pooling, and fully connected layers—and then returns a classification label for the input image. Figure 3 shows the proposed CNN model architecture and the model is described by the following steps:

1. All grayscale images of the dataset [24] are resized into 32×32×1, where 32×32 is an image size and 1 is the grayscale image channel.
2. All DL models accept data as numbers; the Python library is utilized to generate a matrix and array of images.
3. The architecture is a 2-layer deep CNN model: the first Conv layer has 32 filters of 4×4, followed by a Max-pooling layer of 2×2; then the second Conv layer has 64 filters of 3×3, followed by a Max-pooling layer of 2×2, and two FC layers are applied. Malware images are 32×32 in size, and since they are convolved with 32 filters that are each 4×4, the output size of the convolutional layer is (32-4+1) × (32-4+1) that is 29×29.
4. A categorical cross-entropy loss function is employed because the 25 real malware categories/labels are one-hot encoded. The loss function seeks to improve the model by minimizing loss. The used function is defined as

$$Loss = -\sum_{i=1}^{no.of\ output\ class} true_i \log(pred_i) \quad (2)$$

where $true_i$ represents the i^{th} class's actual class label, $pred_i$ represents the i^{th} class's measured probability, and number of output class represents the total number of classes (in this case n is 25) and base 2 is used to calculate the log function.

5. Rectified linear unit (ReLU) as an activation function to incorporate nonlinearities (for neurons).
6. Adam optimizer is used for optimization tasks for multi-class classification problems.
7. The output layer has 25 neurons that match the dataset's 25 malware families, and the output layer has a class probability of the malware samples it belongs to.

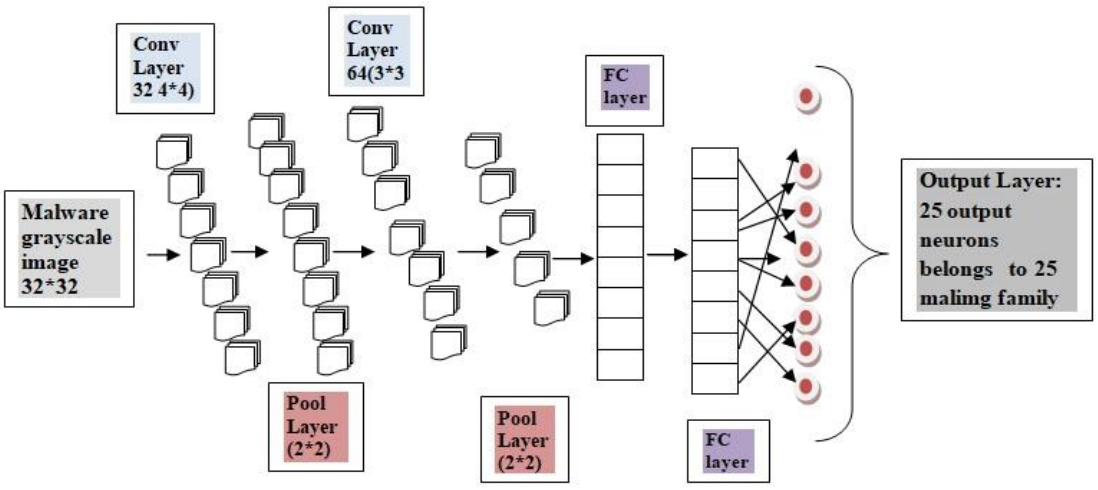

Fig. 3. The proposed CNN model architecture

4. Experiment Results and Execution Environments

All experiments in this study were conducted on a laptop with a 64-bit Windows operating system, an Intel Core i3, 2.40 GHz processor, and 4 GB of RAM. The Python programming language, Python packages, libraries, and the Keras library are mainly used for designing, training, and testing the proposed CNN model. The developed CNN model was

trained on 80% of the sample (7564 malware samples), validated on 10% (841 samples), and tested on 10% (934 samples) of the entire dataset. Accuracy, precision, recall, and F1-score performance measures are used to assess the classification model's accuracy. The proposed CNN architecture was trained and tested, and the test accuracy of 98.179% was attained. Table 2 The CNN model's hyperparameter settings and the results of the experiments presents the different measures of the model with hyperparameter values.

Table 2. The CNN model's hyperparameter settings and the results of the experiments

Parameter	**values**
Image size	32×32 grayscale image
Classification type	Multi-class classification
Train-Validation-Test ratio	80%-10%-10 %
Optimizer	Adam optimizer
Learning rate	0.001
Batch size	25
Epochs	50
Accuracy	98.179%
Precision	0.9739
Recall	0.9817
F1-score	0.9770
Test loss	0.076
Test time	3s (3 ms/step)

Table 3 represents the classification report for the individual malware family; the report clearly illustrates the different performance measurement parameters for each family.

Table 3. Classification Report of CNN model for each dataset class

Malware family	**Precision**	**Recall**	**F1-score**	**Support**
Allaple.A	1.00	1.00	1.00	11
Allaple.L	1.00	1.00	1.00	9
Yuner.A	1.00	1.00	1.00	296
Instantaccess	1.00	1.00	1.00	173
VB.AT	1.00	1.00	1.00	20
Fakerean	0.00	0.00	0.00	9
Lolyda.AA 1	0.93	0.93	0.93	14
C2Lop.gen!G	0.95	0.95	0.95	21
Alueron.gen!J	1.00	1.00	1.00	16
Lolyda.AA 2	1.00	1.00	1.00	20
Dialplatform.B	1.00	1.00	1.00	37
Dontovo.A	1.00	1.00	1.00	27
Lolyda.AT	1.00	1.00	1.00	21
Rbot!gen	1.00	1.00	1.00	21
C2Lop.P	1.00	1.00	1.00	12
Obfuscator.AD	1.00	1.00	1.00	16
Malex.gen!J	1.00	1.00	1.00	11
Swizzor.gen!I	1.00	1.00	1.00	7
Swizzor.gen!E	1.00	1.00	1.00	19
Lolyda.AA 3	1.00	1.00	1.00	7
Adialer.C	0.75	0.92	0.83	13
Agent.FYI	1.00	0.64	0.78	11
Autorun.K	1.00	1.00	1.00	36
Wintrim.BX	1.00	1.00	1.00	11
Skintrim.N	0.91	1.00	0.96	96

Figure 4 depicts the losses in training and validation. The loss on the training and test data reduces quickly and at the same rate, as shown in the diagram, for up to 7 epochs before practically flattening out for the following epochs, indicating that the model generalizes to new data successfully.

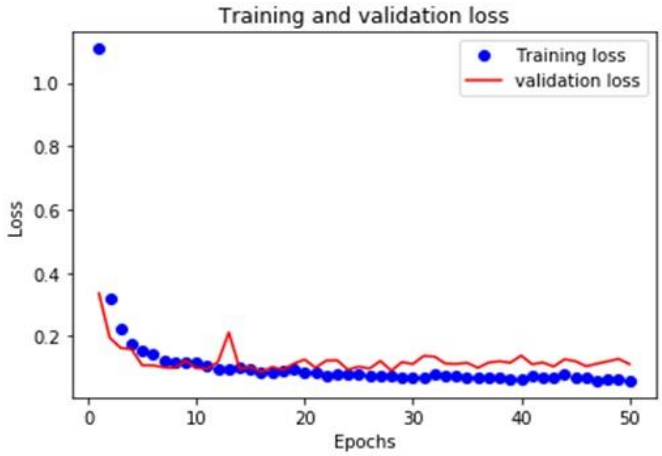

Fig. 4. Plot: Training and validation losses versus epochs

Figure 5 depicts the correctness of the developed model CNN model during training through Epochs. The accuracy versus epoch curve shows that while it initially rises, it eventually approaches a plateau and shows very few deviations (after 18 epochs), which suggests it is no longer able to learn.

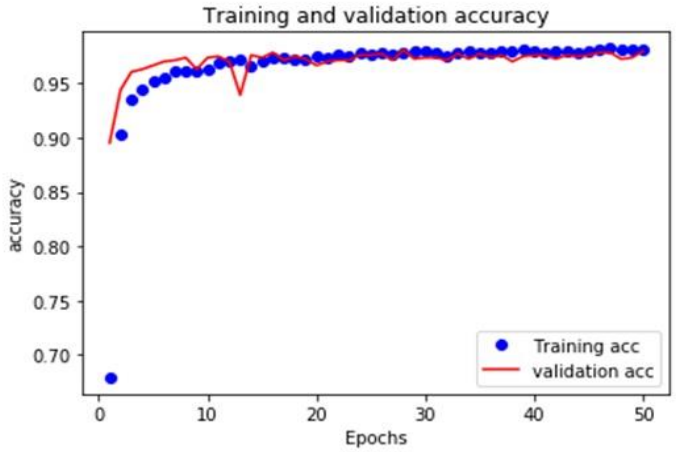

Fig. 5. Plot: Training and validation accuracy versus epochs

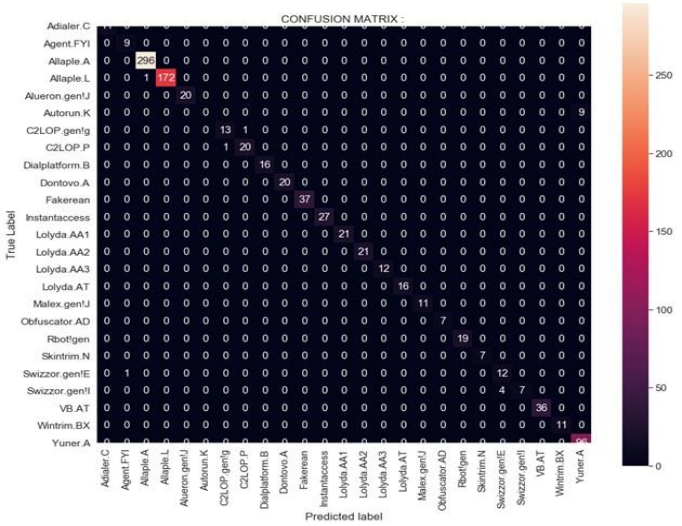

Fig. 6. The confusion matrix of the CNN model

The effectiveness of the designed CNN model is evaluated using a confusion matrix and the CNN model's confusion matrix is depicted in Figure 6 and it clearly illustrates the variants which are correctly classified and variants that are misclassified in different malware families. The confusion matrix shows how some malware families confuse

the training process of DL models (with similar visual features and family structure) and degrade performance. The most perplexing similarity exists between the Autorun.K and Yuner.A malware families (all Autorun.K malware samples were mistakenly classed as Yuner.A) and some samples of Swizzor.gen!I into Swizzor.gen!E. 17 malware samples out of 934 unseen malware test images were mislabeled.

Table 4 compares the proposed model's results to some benchmark traditional methods of malware classification using DL and visualization with the same dataset.

Table 4. Comparison of accuracy for malimg dataset

Author	**Method**	**Accuracy**
Agarap,2017 [6]	CNN-SVM	77.23%
Lu and Li, 2019 [21]	GAN	84%
Kosmidis and Kalloniatis, 2017 [9]	MLP	87%
Mourtaji et al., 2019 [12]	GIST + SVM	92.72%
Kalash et al., 2018 [11]	GIST + SVM	93.23%
Cui et al., 2018 [23]	CNN	94.50%
Bhodia et al., 2019 [25]	ResNet50	94.80%
Vinayakumar et al., 2019 [10]	CNN	96.3%
Jain et al., 2020 [14]	CNN ELM	96.3%, 97.7%
Mourtaji et al., 2019 [12]	CNN	97.02%
Nataraj et al., 2011 [8]	K-NN	97.18%
Sharma et al., 2020 [26]	CNN	97.58%
Kabanga and Kim, 2017 [16]	CNN	98%
Kumar et al., 2018 [13]	CNN	98%
Yajamanam et al., 2018 [27]	GIST+CNN	98%
Lad et al., 2020 [7]	CNN	98.03%
This study	CNN	98.179%
Hamad, 2019 [2]	CNN	98.18%
Kalash et al., 2018 [11]	CNN	98.52%
Yue, 2021 [22]	VGG-19	98.63%
Vasan et al., 2020 [18]	VGG16, ResNeT50	99.50%
Mitsuhashi and Shinagawa, 2020 [17]	VGG-19	99.72%

The designed CNN model has excellent results for each performance metric, as shown in Table 2. It can be observed from Table 3 that the proposed DL architecture offers better performance than the other classic and novel DL based solutions [14,16,21,23] and at the same time some of the studies [17,18] shown better results than the proposed approach because the authors utilized the pre-trained DL models [20,22,25] and some authors [2,11,23] used very high dimensional grayscale images and some [2,18,25] utilized color images versus grayscale images (the results proved that color images with CNN model are better than grayscale images) and some also solved the dataset imbalance issue [21,22,23] and then classified.

Pre-trained models are built very deeply into the architecture, so they increase training and testing time, and for high-dimensional and color images, processing time also increases. The proposed solution has a simple DL architecture, processes, low-dimensional grayscale images, tried to train with small data, the first-time 4×4 filters are applied, and the results are best under this image dimension to the best of our knowledge. This paper includes a comprehensive overview of the dataset used, clearly explains the preprocessing steps, and focuses on multi-class classification [7,9] which is more complex than binary classification [13, 22] and finally, various assessment measures were used to assess the implemented classifiers. In comparison to previous methods suggested in the literature, the proposed model's low complexity—due to low-dimensional images and a simple DL-based architecture—exhibits fast response, quick classification, a high classification rate, and limited processing resources.

5. Conclusion and Future Work

Malware classification is both an important and a severe research problem. A staggering increase in malware and attack variants constantly poses a security risk to the cyber world. The primary purpose and contribution of this paper is to enhance classification accuracy by implementing the DL model that takes advantage of CNN to classify malware images. This study is significant and the proposed CNN model proves to be very promising as it enhances the performance and achieves better accuracy, good test time for more challenging and highly imbalanced datasets. The paper's main contribution is to propose and enhance the malware classification and the experimental results show that the proposed model has performed well and demonstrate the viability of the proposed approach. Malware can be automatically classified using its visual representation, which offers several advantages compared to more traditional techniques. The advantage of this strategy is the ease with which new variants of well-known malware families can be classified. In such a situation, this method can come in handy. In the future, this work will extend to the transformation of malware into color malware images and the implementation of other DL models needs to be investigated to make

classification more time-efficient and accurate. The suggested CNN model needs to be assessed with new datasets, primarily malware datasets for IoT networks, due to its low complexity. Although, the proposed solution used spatial features for the malware classification; it also investigates how the temporal features in the malware images would be used for classification, which would be a good topic for future research.

References

[1] Yadav B., Tokekar S. (2021) 'Deep Learning in Malware Identification and Classification', In: Stamp M., Alazab M., Shalaginov A. (eds) Malware Analysis Using Artificial Intelligence and Deep Learning. Springer, Cham, pp. 163-205. https://doi.org/10.1007/978-3-030-62582-5_6

[2] Hamad,N. (2019)'Detection of Malicious Activities in Internet of Things Environment Based on Binary Visualization and Machine Intelligence',Wireless Personal Communications, Vol. 108, pp.2609-2629. https://doi.org/10.1007/s11277-019- 06540-6

[3] Singh, A., Handa, A., Kumar, N. and Shukla, S.K. (2019) 'Malware classification using image representation', In International Symposium on Cyber Security Cryptography and Machine Learning, pp. 75-92. https://doi.org/10.1007/978-3-030-20951-3_6

[4] Malware statistics and Trends Report (2020) [online] by AV-test institute.https://www.av-test.org/en/statistics/malware/(Accessed 25 January 2021).

[5] McAfee labs threats report [online] November 2020. https://www.mcafee.com/enterprise/enus/assets/reports/rpquarterly-threats-nov2020.pdf. (Accessed 25 January 2021).

[6] Agarap, A.F. (2017) 'Towards building an intelligent anti-malware system: a deep learning approach using support vector machine (SVM) for malware classification', ArXivpreprint, arXiv: 1801.00318.

[7] Lad, S.S. and Adamuthe, A.C. (2020)'Malware Classification with Improved Convolutional Neural Network Model', International Journal of Computer Network & Information Security, Vol. 12, No. 6, pp. 30- 43. https://doi.org/10.5815/ijcnis.2020.06.03

[8] Nataraj, L., Karthikeyan, S., Jacob, G. and Manjunath, B.S. (2011)'Malware images: visualization and automatic classification', In Proceedings of the 8th international symposium on visualization for cyber security, pp. 1- 7. https://doi.org/10.1145/2016904.2016908

[9] Kosmidis, K.and Kalloniatis, C. (2017) 'Machine Learning and Images for Malware Detection and Classification',21st Pan-Hellenic Conference on Informatics, pp. 1– 6. https://doi.org/10.1145/3139367.3139400

[10] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P. and Venkatraman, S. (2019) 'Robust intelligent malware detection using deep learning', IEEE Access, Vol. 7, pp.46717-46738. https://doi.org/ 10.1109/ACCESS.2019.2906934

[11] Kalash, M., Rochan, M., Mohammed, N., Bruce, N.D., Wang, Y. and Iqbal, F. (2018) 'Malware classification with deep convolutional neural networks', In 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1-5. https://doi.org/10.1109/NTMS.2018.8328749

[12] Mourtaji, Y., Bouhorma, M. and Alghazzawi, D. (2019) 'Intelligent framework for malware detection with convolutional neural network', In Proceedings of the 2nd International Conference on Networking, Information Systems & Security, pp. 1- 6. https://doi.org/10.1145/3320326.3320333

[13] Kumar, R., Xiaosong, Z., Khan, R.U., Ahad, I. and Kumar, J. (2018) 'Malicious code detection based on image processing using deep learning', In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, pp. 81- 85. https://doi.org/10.1145/3194452.3194459

[14] Jain, M., Andreopoulos, W. and Stamp, M. (2020) 'Convolutional neural networks and extreme learning machines for malware classification', Journal of Computer Virology and Hacking Techniques, Vol. 16, No. 3, pp.229- 244. https://doi.org/10.1007/s11416-020-00354-y

[15] Ansari, M.A. and Singh, D.K. (2021) 'Monitoring social distancing through human detection for preventing/reducing COVID spread', International Journal of Information Technology, Vol. 13, No. 3, pp.1255-1264. https://doi.org/10.1007/s41870-021-00658-2

[16] Kabanga, E.K. and Kim, C.H. (2017) 'Malware images classification using convolutional neural network', Journal of Computer and Communications, Vol. 6, No. 1, pp.153- 158. https://doi.org/10.4236/jcc.2018.61016

[17] Mitsuhashi, R. and Shinagawa, T. (2020) 'High-accuracy malware classification with a malware-optimized deep learning model', ArXiv preprint,arXiv: 2004.05258.

[18] Vasan, D., Alazab, M., Wassan, S., Safaei, B. and Zheng, Q. (2020) 'Image-Based malware classification using ensemble of CNN architectures(IMCEC)', Computers & Security, Vol. 92, pp.101748. https://doi.org/10.1016/j.cose.2020.101748

[19] Gavriluţ, D., Cimpoeşu, M., Anton, D. and Ciortuz, L. (2009) 'Malware detection using machine learning', In International Multiconference on Computer Science and Information Technology, pp. 735-741. https://doi.org/10.1109/IMCSIT.2009.5352759

[20] Simonyan, K. and Zisserman, A. (2014) 'Very deep convolutional networks for largescale image recognition', ArXiv preprint,arXiv: 1409.1556.

[21] Lu, Y. and Li, J. (2019) 'Generative adversarial network for improving deep learning based malware classification', In Winter Simulation Conference (WSC), pp. 584- 593. https://doi.org/10.1109/WSC40007.2019.9004932

[22] Yue, S. (2017) 'Imbalanced malware images classification: a cnn based approach', ArXiv preprint, arXiv: 1708.08042.

[23] Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G.G. and Chen, J. (2018) 'Detection of malicious code variants based on deep learning', IEEE Transactions on Industrial Informatics, Vol. 14, No. 7, pp.3187- 3196. https://doi.org/10.1109/TII.2018.2822680

[24] Malimg dataset (2011) Based on grayscale images [online].https://www.kaggle.com/afagarap/malimg-dataset. (Accessed 25 January 2021).

[25] Bhodia, N., Prajapati, P., Di Troia, F. and Stamp, M. (2019) 'Transfer learning for imagebased malware classification', ArXiv preprint,arXiv: 1903.11551. https://doi.org/10.5220/0007701407190726

[26] Sharma, G.A., Singh, K.J. and Singh, M.D. (2020) 'A deep learning approach to imagebased malware analysis', Progress inComputing, Analytics and Networking. AISC, pp.327-339. https://doi.org/10.1007/978-981-15-2414-1_33
[27] Yajamanam, S., Selvin, V.R.S., Di Troia, F. and Stamp, M. (2018) 'Deep Learning versus Gist Descriptors for Image-based Malware Classification', In Icissp, pp. 553-561. https://doi.org/10.5220/0006685805530561

Authors' Profiles

Balram Yadav was born in Rae Bareli (UP), India, in 1984. He received his B.E. (Computer Science) and M.Tech. (Information Technology) degree from Mahakal Institute of Technology, Ujjain (M.P.) India in 2006 and 2011 respectively. Presently, he is working as an Assistant Professor for the Computer Engineering Department of Mahakal Institute of Technology, Ujjain (M.P.) India. He has more than 13 years of teaching experience. His teaching and research interests include Malware analysis, Malware detection, and classification; Image processing, Theory of Computation, Analysis and Design of algorithms, Machine and Deep learning.

Dr. Sanjiv Tokekar received his Bachelor of Engineering, (Electronics), Master of Engineering (Applied Electronics), and Ph.D., (Electronics Engineering) from Devi Ahilya Vishwavidyalaya, Indore, India in the years 1982, 85 and 96, respectively. Currently, he is working as a Professor, Director, and Head of the Department of Electronics and Telecommunication in the Institute of Engineering and Technology, DAVV Indore, MP, India. More than 60 research papers (in International Journal, national journal, International conferences, and national conferences) are there to his credit. He is a senior member of IEEE, a member of the Computer Society of India, Indian Society of Technical Education. He is also chair of the IEEE MP Subsection under the Bombay section. His teaching and research areas are VLSI, Computer Networks, Telecommunication Networks, Computer Architecture, Digital Signal Processing, and Performance Evaluation of Computer Systems.

How to cite this paper: Balram Yadav, Sanjiv Tokekar, "Malware Multi-Class Classification based on Malware Visualization using a Convolutional Neural Network Model", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.15, No.2, pp. 20-29, 2023. DOI:10.5815/ijieeb.2023.02.03