

# Intelligent malware classification based on network traffic and data augmentation techniques

Ammar D. Jasim<sup>1</sup>, Rawaa Ismael Farhan<sup>2</sup>

<sup>1</sup>Department of Information and Communication Engineering, College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

<sup>2</sup>Department of Basic Science, College of Dentistry, University of Wasit, Wasit, Iraq

## Article Info

### Article history:

Received Aug 24, 2022

Revised Jan 6, 2023

Accepted Jan 10, 2023

### Keywords:

Convolutional neural network

Data augmentation

Deep learning

Gray scale image

Malware

## ABSTRACT

To prevent detection, attackers frequently design systems to rearrange and rewrite their malware automatically. The majority of machine learning techniques are not sufficiently resistant to such re-orderings because they develop a classifier based on a manually created feature vector. Deep learning techniques like convolutional neural networks (CNN) have lately proven to perform better than more traditional learning algorithms, especially in applications like picture categorization. As a result of this success, CNN network proposed with data augmentation techniques (to enhance the performance) to classify malware samples. We trained a CNN to classify the photos using converted grayscale images from malware files. Our methodology outperforms other methods with an accuracy of 98.80%, according to experimental results.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Rawaa Ismael Farhan

Department of Basic Science, College of Dentistry, University of Wasit

Wasit, Iraq

Email: ralrikabi@uowasit.edu.iq

## 1. INTRODUCTION

Malicious software, or malware, represents a serious issue for contemporary devices of computing, such as mobile phones and computers. While professional anti-virus technologies try to identify and repair an infected device, attackers are motivated by financial gain to add deft redundancies to programming to reinfect the computer or mobile phone rapidly and automatically. The majority of new malware are variations of existing malware due to on each contaminated computer there are different characteristics of malware. When malware spread from computer to another uses engines of mutation to create a different hash representation of its executable file, according to earlier studies on the classification of malware [1]-[5]. Malware samples often belong to a family that has common behaviors. As a result, the idea of developing a mechanism that can effectively categorize malware according to its family, regardless of whether it is a variety, seems extremely beneficial and a way to deal with the virus's explosive expansion. In contrast to conventional approaches, we adopt a completely different strategy in this study to evaluate and categorize malware. To solve this issue, we employ a convolutional neural network (CNN) for malware family's detection by enhanced deep learning architecture with data augmentation techniques. The potential of implementing CNNs, though, has not been thoroughly investigated in many other domains. Cyber security is one industry that could gain a lot from deep learning developments. Given the recent advancement of deep learning (especially CNNs) inside a range of classification tasks. To classify malware into families, the malware's executable files are used, where the deep learning model learns the visual features resulting from converting executable files into grayscale images, and thus the difference is detected in the samples, although the general shape is preserved, it can notice minor changes. We estimate it is feasible to identify malware with much more precision than any simple

learning technique, such as support vector machines (SVM) [6]-[10]. CNNs have succeeded particularly well in issues involving pictures. Inspired by this achievement, we convert the malware categorization issue into an image. Using CNNs, a categorization issue will be handled. We present each malicious binary file as a grayscale image in accordance with earlier work [11]-[14], and develop a classification CNN architecture. It should be noted that prior research [15]-[19] demonstrated that malicious from similar family is looks the same, which is advantageous in terms of a CNN's ability to detect related patterns. Special important given because different malware variants are typically created using the same, or code that is quite close to it.

## 2. METHOD

Data diversity is the key problem of malware detection. To overcome this problem, data augmentation represents an artificial strategy to increase the input instances variety in the training phase, thus, no new instances collecting of really. The main contribution is to develop an enhanced model of data augmentation for malware family classification by malware variants augmentation, then exploits the CNN network to enhance the classification of images.

### 2.1. Malware files to gray-scale image

To create new malware, malware developers often alter a small portion of the code that was already present [20]. These small adjustments are simple to track if malware is visualized as a picture. This and earlier research [21] served as inspiration for how we represent malware binary files as gray-scale graphics. The procedure for converting malware binary data to grayscale images is shown in Figure 1.

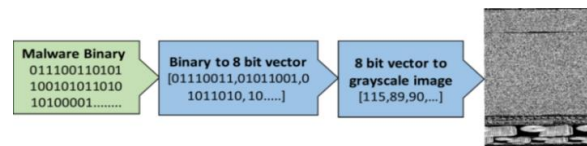


Figure 1. Visualize binary file to grayscale image

A vector of 8-bit unsigned integers containing a particular malicious binary file is initially read. The binary values of each component in this array are translated to their associated integer file numbers to make a new integer array that describes the malware sample. After that, a two-dimensional matrix representing the output integer array is created and displayed. Malware variations from the same family typically have a strong similarity. Gray scale images of malware belonging to different malware families are shown in Figure 2. Figure 2(a) is a multi-threaded, polymorphic network worm capable of spreading to other computers connected to a local area network (LAN) and performing denial-of-service (DoS) attacks against targeted remote Web sites. Figure 2(b) is a trojan that attempts to modify DNS settings on network routers. Figure 2(c) this threat can perform a number of actions of a malicious hacker's choice on your PC. Figure 2(d) Such ransomware are a form of malware that is specified by online frauds to demand paying the ransom by a sufferer. Figure 2(e) will certainly advise its victims to launch funds move for the purpose of reducing the effects of the modifications that the Trojan infection has actually introduced to the sufferer's gadget. Figure 2(f) worms automatically spread to other PCs. They can do this in a number of ways, including by copying themselves to removable drives, network folders, or spreading through email.

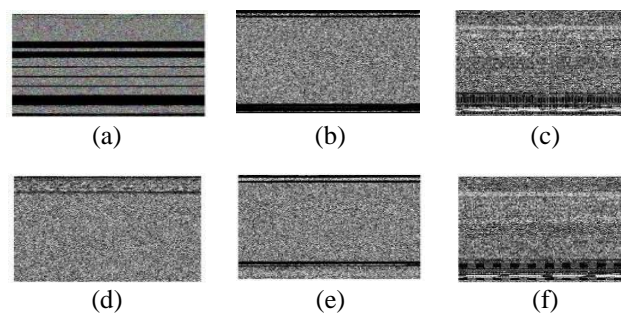


Figure 2. Gray scale images of different malware families, (a) Net-Worm: W32/Allapple.A, (b) Win32/Alureon.gen!J, (c) Trojan: Win32/Autorun.K, (d) Ransomware: Malex.gen!J, (e) Win32/Obfuscator.ACY, and (f) Worm: Win32/Yuner.A

## 2.2. Data augmentation

Data augmentation is a method for enhancing datasets artificially. Collect samples from the dataset, make some changes to it, and then add them to the original dataset. As a result, your dataset has grown by one sample as shown in Figure 3. One of the most crucial aspects of deep computer vision is data augmentation. You should always enhance your data when training your neural network, like, always. Otherwise, your model does not perform as well as it should since your dataset is not being used efficiently.

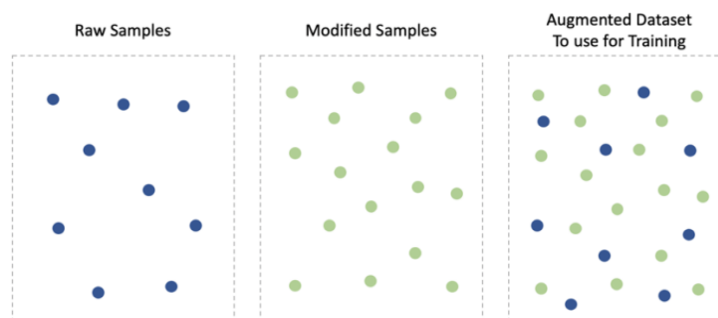


Figure 3. Data augmentation concept

Due to limited data that leads to overfitting problem in deep learning models, data augmentation techniques are relied upon. There are several geometrical transformations of data augmentation:

- Flipping: the x-axis is changed much more typically than the y-axis. This augmentation is one of the simplest to use and has been useful when applied to datasets. this does not keep the transformation of the label.
- Color space: digital picture data is often stored as a matrix of the dimensions (height, width, and color channels). Another highly useful tactic is to perform modifications in the pixel intensity region. Isolating a specific color channel, such as red (R), green (G), or blue (B), is a relatively easy way to improve color. By separating one matrix and connecting zeros matrices from another color channels, a picture can be swiftly transformed into its equivalent inside one color channel. Additionally, the intensity of the image can be readily changed by altering the Three colors using basic matrix operations. Make a color histogram of the image for more intricate color adjustments. These histograms' intensity values can be changed to control the lighting, much to how image editing software does it.
- Rotate: the picture is turned to a right or left to an angle between  $1^\circ$  and  $359^\circ$  to perform out rotation augmentations. The rotation degree parameter has a considerable effect on the security of rotation augmentations. minor rotations between  $-1$  to  $-20$  or  $1$  and  $20$  could be useful on tasks of number recognition as modified national institute of standards and technology database (MNIST), but with the increasing on rotation degree, the data label is no longer kept after transformation.
- Noise injection: the "noise insertion" procedure comprises introducing an array of real numbers that are periodically chosen using a Gaussian distribution. The method of noise injection refers to adding "noise" artificially to the CNN input data during the training process. In the feature space during training, making it difficult for the CNN to find a solution that fits precisely to the original training dataset, thereby reducing the overfitting of the CNN. Thus, CNN's learn more strong features by adding noise to images.

## 2.3. Enhanced CNN-based malware detection

In this section describe the classification of malware samples based on their family using CNN model. A feed-forward neural network with biological inspiration specifically, the way the human visual system is organized is known as a convolution neural network (CNN) [22], [23]. The most advanced deep learning models for image categorization now is CNN. The neurons that make up CNN have biases and weights that can be learned. These three elements makeup the majority of CNNs [24]:

- Convolutional layer: this layer sequentially performs a series of convolution operations on a picture. Typically, these filters take information from the input image about edges, colors, and shapes. In essence, the filters work on the subregions of a picture and execute computations so that they output a single value for each subregion.
- Pool layer: this layer is in charge of downsampling (dimensionality reduction) the data obtained from convolution layers in order to decrease the processing time and enable the scale of the data to be handled by the computational resources. This is because as a result of pooling, there are fewer parameters that can

be learned in the network's deeper layers. A frequent pooling technique is max pooling, which maintains the maximum value in a zone.

- Full connected layer: used to classify the output produced by the pooling and convolution layers. Each neuron in this layer is linked to every neuron in the layer above it.

The hyperparameters of CNN architecture were selected by a grid search which contained the learning rate and the number of layers (convolutional and fully-connected). The executable file of malware is represented as a grayscale image then provided as input for the CNN network. First, images should be rescaled to feed the CNN model. Thus, the resulting images size saves the computational resource usage to provide the best accuracy. Secondly, the imgaug Python library has been used for image augmentation by using (Gaussian, Laplacian, and Poisson) as additive noise methods.

Our CNN architecture described in Figure 4. contained one input layer and three convolution layers each one as (convolution, activation, pooling) to learn hierarchical features and one fully-connected layer. The rectified linear unit (ReLU) activation function in the three convolution layers was used to signal the distinct specification of potential features. While, the spatial size of the features is reduced by pooling operation, thus applying certain order of robustness versus noise and distortion.

The fully-connected layer contained 256 neurons. The output layer provides the output which represents the malware class based on their family by using the softmax function. To overcome the overfitting problem dropout has been employed for normalization. As result, through forward propagation, it drops randomly a proportion of neurons to avoid the dependencies between neurons. Figure 4 show the enhanced CNN architecture.

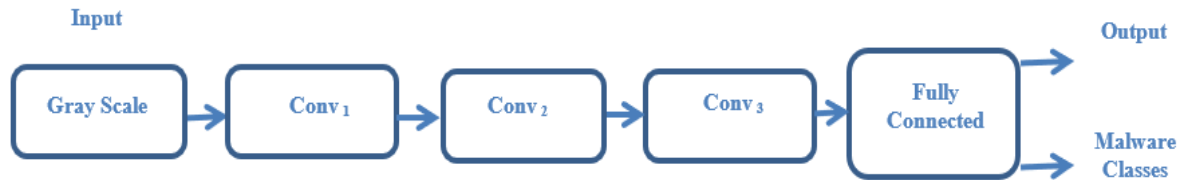


Figure 4. Enhanced CNN architecture

### 3. DATASET

There are 9,339 malware samples in this collection [25] that are shown as grayscale images in total. One of the 25 malware families/classes is represented by each sample of malware in the collection. Additionally, the dataset's distribution of samples from a given malware family varies. In our tests, we choose at random 0.90 of the malware samples in a family for training and the other 0.10 for testing. Finally, we have 8,394 malware training samples and 945 testing samples.

### 4. RESULTS

Accuracy depicts the performance of the detection method as shown in Table 1. In contrast to earlier work, we suggested a 2D-CNN using a data augmentation technique to maximize the use of the dataset. In earlier research, the GIST + SVM classifier was first trained after several malware file-related attributes were chosen. Second, it was suggested to train the 2D-CNN so that it could independently learn features from images. Our solution uses data augmentation techniques with 10 epochs, outperforming the performance of these two methods (less than that in previous work). In this manner, we successfully utilized the dataset and avoided overfitting. Figure 5 shows the metric values of the proposed model and according to shown graph our suggested method outperforms other architectures.

Table 1. Dataset's results

Method	Accuracy of method
Method in [2]	97.18%
GIST + SVM [25]	93.23%
2D-CNN [25]	98.52%
Our method	98.80%

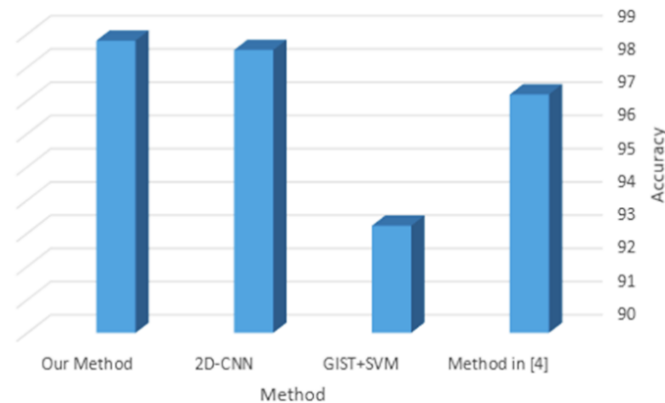


Figure 5. Comparison between our method and relevant methods in terms of accuracy

## 5. CONCLUSION

The security danger posed by malware to computer systems is growing. In order to develop efficient ways to resist malware attacks, it is important to evaluate malware behavior and categorize samples. The aim of this paper is to malware family's detection in a metamorphic environment using an enhanced CNN model by data augmentation. Because the standard CNN classification performance of the model is low, so it is not appropriate for the detection and classification of malware. To evaluate the performance of the learning algorithm the augmentation has been measured with different noise ratios. It is clear from the results that data augmentation affects the performance of malware family classification based on malware gray scale images. Our model is able to classify malware samples with 98.80 accuracy, thus outperforming on other approaches in literature.





## REFERENCES

- [1] L. Nataraj, S. Karthikeyan, and B. S. Manjunath, "SATTVA: SpArSiTy inspired classification of malware variants," in *IH and MMSec 2015 - Proceedings of the 2015 ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2015, pp. 135–140, doi: 10.1145/2756601.2756616.
- [2] F. O. Catak, J. Ahmed, K. Sahinbas, and Z. H. Khand, "Data augmentation based malware detection using convolutional neural networks," *PeerJ Computer Science*, vol. 7, pp. 1–26, Jan. 2021, doi: 10.7717/PEERJ-CS.346.
- [3] N. Fleury, T. Dubrunquez, and I. Alouani, "PDF-malware: An overview on threats, detection and evasion attacks," *arxiv preprints*, Jul. 2021, [Online]. Available: <http://arxiv.org/abs/2107.12873>.
- [4] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial machine learning attacks and defense methods in the cyber security domain," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–36, Jun. 2021, doi: 10.1145/3453158.
- [5] Y. Huang, V. De Bortoli, F. Zhou, and J. Gilles, "Review of wavelet-based unsupervised texture segmentation, advantage of adaptive wavelets," *IET Image Processing*, vol. 12, no. 9, pp. 1626–1638, Sep. 2018, doi: 10.1049/iet-ipr.2017.1005.
- [6] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "DL 4 MD: A deep learning framework for intelligent malware detection," in *Proceedings of the International Conference on Data Science (ICDATA)*, 2016, pp. 61–67.
- [7] X. Li, J. Liu, Y. Huo, R. Zhang, and Y. Yao, "An android malware detection method based on AndroidManifest file," in *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Aug. 2016, pp. 239–243, doi: 10.1109/CCIS.2016.7790261.
- [8] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *Journal of Network and Computer Applications*, vol. 153, p. 102526, Mar. 2020, doi: 10.1016/j.jnca.2019.102526.
- [9] M.-Y. Su, J.-Y. Chang, and K.-T. Fung, "Machine learning on merging static and dynamic features to identify malicious mobile apps," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, Jul. 2017, pp. 863–867, doi: 10.1109/ICUFN.2017.7993923.
- [10] N. Šrđić and P. Laskov, "Hidost: a static machine-learning-based detector of malicious files," *Eurasip Journal on Information Security*, vol. 2016, no. 1, p. 22, Dec. 2016, doi: 10.1186/s13635-016-0045-0.
- [11] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security - VizSec '11*, 2011, pp. 1–7, doi: 10.1145/2016904.2016908.
- [12] O. P. Samantray, S. N. Tripathy, and S. K. Das, "A theoretical feature-wise study of malware detection techniques," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 12, pp. 879–887, Dec. 2018, doi: 10.26438/ijcse/v6i12.879887.
- [13] J. Park and H. Kim, "k-Depth mimicry attack to secretly embed shellcode into PDF Files," in *Lecture Notes in Electrical Engineering*, vol. 424, 2017, pp. 388–395.
- [14] M. Ozsoy, C. Donovick, I. Gorelik, N. Abu-Ghazaleh, and D. Ponomarev, "Malware-aware processors: A framework for efficient online malware detection," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, Feb. 2015, pp. 651–661, doi: 10.1109/HPCA.2015.7056070.
- [15] Y. Qiao, W. Zhang, Z. Tian, L. T. Yang, Y. Liu, and M. Alazab, "Adversarial ELF malware detection method using model interpretation," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 605–615, Jan. 2023, doi: 10.1109/TII.2022.3192901.





- [16] G. Martin *et al.*, “Mobile malware detection using consortium blockchain,” *Advances in Information Security*, vol. 54, pp. 137–160, 2022.
- [17] G. Iadarola, F. Martinelli, A. Santone, and F. Mercaldo, “Assessing the robustness of an image-based malware classifier with small level perturbations techniques,” in *Advances in Information Security*, vol. 54, 2022, pp. 69–84.
- [18] N. Penning, M. Hoffman, J. Nikolai, and Y. Wang, “Mobile malware security challenges and cloud-based detection,” in *2014 International Conference on Collaboration Technologies and Systems (CTS)*, May 2014, pp. 181–188, doi: 10.1109/CTS.2014.6867562.
- [19] O. Aslan and A. A. Yilmaz, “A new malware classification framework based on deep learning algorithms,” *IEEE Access*, vol. 9, pp. 87936–87951, 2021, doi: 10.1109/ACCESS.2021.3089586.
- [20] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, “Local binary patterns for still images,” in *Computer Vision Using Local Binary Patterns*, Computatio., London: Springer, 2011, pp. 13–47.
- [21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6314 LNCS, no. PART 4, 2010, pp. 778–792, doi: 10.1007/978-3-642-15561-1\_56.
- [22] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, “Malware classification with deep convolutional neural networks,” in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Feb. 2018, vol. 2018-Janua, pp. 1–5, doi: 10.1109/NTMS.2018.8328749.
- [23] Keras Team, “Keras Documentation,” keras.io. <https://keras.io> (accessed Dec. 20, 2022).
- [24] “Intro to Convolutional Neural Networks.” tensorflow.org. <https://www.tensorflow.org/tutorials/images/cnn> (accessed Dec. 23, 2022).
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

## BIOGRAPHIES OF AUTHORS



**Assist. Prof. Dr. Ammar D. Jasim**     is Assistant Professor at college of Information Engineering, Al-Nahrain University, Iraq. He Holds a Ph.D. degree in Information Engineering with specialization in networks. His research areas are Networks, Communications, Security, Artificial Intelligence. He is head of information system department in Al-Nahrain University, Iraq. He is supervisor for a number of students in Ph.D. He can be contacted at email: ammar@coie.nahrain.edu.iq.



**Rawaa Ismael Farhan**     is Assistant Professor at college of Computer Science and Information Technology, Wasit University, Iraq. She was received M.Sc. with specialization of networks in Computer Science from Osmania University, Heyderabad, India in 2014. She Holds a PhD degree in Network Security from Technology University, Iraq. Her 15 researchs areas are Networks, Security, Artificial Intelligence, IoT, Cloud Computing. She can be contacted at email: ralrikabi@uowasit.edu.iq.