2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Benghazi, Libya 21-23 May 2023.

Malware Analysis in Cyber Security based on Deep Learning; Recognition and Classification

1st Mohamed Elalem
Electrical and Computer Dept.
Elmergib University
Alkhoms, Libya
maelalem@elmergib.edu.ly

2nd Tahani Jabir
Electrical and Computer Dept.
Elmergib University
Alkhoms, Libya
tahani.jabir93@gmail.com

2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA) | 979-8-3503-1989-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/MI-STA57575.2023.10169310

***Abstract*—Cyber security in wireless communications can be an unwieldy subject, given the amount of malware that has been increasing rapidly in the last few years. This generates serious security problems for public agencies and the government institutions. In order to mitigate the influence of malware deployment and pervasiveness, new recent identification and classification algorithms for malware that adapt deep learning techniques are studied and figured out based on their features and behaviors. This study introduces a deep learning algorithm to identify different malware families. To implement the proposed approach, malware color-based images (RBG) are used directly. Then these malware images are identified and classified by considering the benefits of leveraged Convolutional Neural Networks (CNNs), which have the ability to automatically extract those features. A challenging malware classification experiment using the MaleVis dataset confirms that the adapted model outperforms better functional classification compared to the traditional machine learning models and achieves very good accuracy based on the MaleVis dataset.**

***Index Terms*—MaleVis malware dataset, classification, convolutional neural networks, deep learning.**

I. Introduction

Malware or malicious software is inquisitive scripts that is launch to damage stored data, computers and network resources, such as worms, trojan horses, backdoors, adware, and spyware [1]. In the Internet age, when the internet has become an integral part of our lives, and especially during the COVID-19 pandemic, so many people work and study at home due to this pandemic. This increases internet usage, entices more people to spend a lot of time online, and opens up more opportunities for cybercrime [2]. Malware scripts are used by legitimate users to steal information, deceive, or make financial promises. These illegal methods are expanding alarmingly each year. SonicWall, an American cyber security company, reports that nearly 10 billion in malware has been identified, with a total number of 5.6 billion attacks in 2020 [3], [4]. Fig. 1 shows the annual number of malware attacks for the last seven years. from 2015 to 2022. Malware attacks started as a hobby for hackers and researchers, and they now happen everywhere in the world. It is currently evolving into a noble global society. Programmers with training are enthusiastic about making money quickly. Hacking applications are now as prevalent and numerous as legitimate software packages, creating a multi-million dollar industry [5]. Furthermore, marauders may be drawn to the availability of customer service and technical support for online communities.

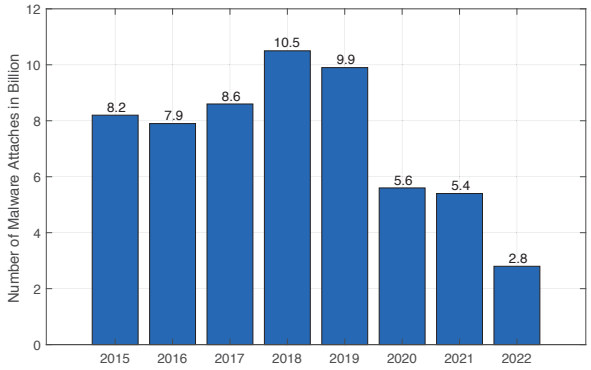

Fig. 1: Malware attacks for the seven last years.

A lot of malware attacks happen which affect all networks, and experiences security incidents at government and public institutions. Malware recognition and classification have become very critical cyber security issues. Although the samples of anti-virus data have massive features, performing the analysis on this data manually is a difficult and tedious process. So these processes should be automated. This will also help reduce the number of files that require manual analysis. Previous studies done on malware detection and classification claimed that malware samples are usually categorized into families sharing common behaviors. The vast majority of new malware is a variation on an already existing behavior [6]–[8]. Therefore, any developed methods for feature analysis might be efficiently used to classify the new malware into the relevant family. This variant of these features is useful because it limits the rapid spread of malware. The essential purpose of this research is to explore different approaches to malware classification and solve this problem applying deep learning techniques. In few last decades, deep learning has become the effective approach to solve engineering problems in various application. it has brought very good performance to a wide variety of tasks in many fields, such as image processing, speech recognition, medical image and information process-

ing, cyber security, *etc* [9]. One area that could benefit from deep learning solution is network cyber security. We believe that the latest success of deep learning (particularly CNN) for various classification problems can classify malware with better performance and accuracy than traditional malware.

Support Vector Machines (SVM) and K-Means Clustering are examples of machine learning algorithms. CNNs are very successful with problems with images and huge amounts of data. Therefore, we work with the malware code classification problem and turn it into an image classification problem and address it with CNN. Based on the previous work [10], we render each malware as a grayscale image and train a CNN for detection and classification. Important improvements and valued results have been achieved from this work compared to previous related works.

In the past, classification techniques basically depended on two types of analysis techniques: static analysis and dynamic analysis [9], [11]. They focus on inspecting the malware code without executing it. In dynamic analysis, the malware is monitored dynamically during the feature execution.

Static analysis provides significant insight into malware identification and classification, but its main weakness lies in managing packing and obfuscation [12]. As a result, dynamic analysis has received a lot of attention recently because it is significantly less susceptible to code obfuscation transformations. However, dynamic analysis has its own set of weaknesses, resulting in an inaccurate picture of malware behavior. Due to the limits of statics and mechanics, an analysis that relies on just one of these is not sufficient to properly classify malware.

Regarding malicious code visualizations, there are many tools that allow us to visualize and analyze binary data such as common text and binary editors. Many previous studies have suggested the use of visual representations of malware [13].

Traditional machine learning techniques suffer from the time required to extract complex image texture features. To meet this challenge, a deep learning approach is used to efficiently recognize and classify images. It has the ability to discover key features and its own pattern set.

After this introduction, the following sections are organized as follows: Section II introduces the importance of malware detection for wireless communication networks and cyber security. Section IV discusses the paper methodology and introduces the dataset used for training phase. Section III briefly explains the convolutional neural network and its layers, including the configuration details of the model used in the present paper. Section V presents the experiments and results, while Section VI draws conclusions and plans for future work.

II. Wireless Communication and Cyber Security

Because so many people rely on wireless systems in their daily lives, the security issue has taken on a significant importance. Due to critical operational constraints, one of the underlying challenges of wireless networks is ensuring security in a communication network. Due to its inadequate security measures, the wireless network is a simple target for malware (worms, viruses, malicious code, etc.) attacks. Malware spreads throughout the wireless network after first infecting a sensor node. Implementing a defense mechanism against malicious software is essential due to the epidemic nature of worm transmission in the network. A deep learning model based on CNN is adopted in this work. The proposed algorithm has the capability to handle a huge number of data (around 12 thousand malware samples), which can identify and classify the malware type faster and automatically. As a result of this model, we reduced the number of infectious lymph nodes while also slowing the spread of malware.

III. Convolutional Neural Networks

In order to distinguish between various objects in input data, such as an image, CNN, a deep learning algorithm, assigns importance weights to each object. Comparatively speaking, CNN needs much less preprocessing than other classification algorithms. The manual development of the filters is necessary for the primitive method, but with sufficient training, the CNN can learn these filters and properties. The CNN's architecture is similar to the neuronal connectivity patterns found in the human brain. By combining those fields, the entire range of neurons is covered [15], [16]. By using the appropriate filters, CNN can detect spatial and temporal dependencies in images. Because there are fewer important parameters and the weights can be reused, the architecture adapts to image data sets better. In other words, the network can be trained to comprehend images' subtleties better.

A. Convolutional Layer(The Kernel)

The convolutional layer is the most important part of the CNN architecture. Traditionally, the function of the first convolution layer is to capture some principle features such as edge, color, and gradient placement. As layers are added, the structure of the layers also adapts to higher-level functions, providing a better understanding of the images in the dataset. A group of convolutional filters, or "kernels," make up the convolutional layer. that are used to extract various features from the given image. The kernel performs the mathematical operation of convolution between input images and a filter of predefined size. The output of this layer is usually passed to a ReLu activation nonlinear function [16], [18].

Convolution operations' primary job is to find high-level features in the input image. CNN isn't required to only use one convolutional layer. The convolution function's dimensionality is initially decreased in comparison to the input; after that, it either increases or stays the same. Applying "Valid Padding" in the first instance and "Same Padding" in the latter.

B. Pooling Layer

Pooling layers, like convolution layers, aid in minimizing the spatial size of the convolution function. Through dimensionality reduction, this lowers the amount of computation needed to process the data. Additionally, it aids in extracting

crucial aspects of rotation and position. This effectively continues the process of model training. The two types of pooling layers are maximum pooling and average pooling. When used as a denoiser, Max Pooling returns the highest value from the area of the image that the kernel has covered. The average values are calculated using the area of the image that the kernel has covered. Simple noise reduction through dimensionality reduction is all that average pooling does. As a result, we can conclude that maximum pooling outperforms average pooling.

C. Fully Connected Layer

Typically, this layer is found A layer known as the Fully Connected Layer (FCL) is located at the bottom of the CNN structure. Each node in this layer is connected to each node in the layer before it. CNN is regarded as its classifier. Being a feed-forward ANN, it operates in a similar manner to a traditional multiple-layer perceptron neural network. from the flattened feature map. A drop-out layer comes after this layer. The dro-pout layer enhances the mode's generalization capabilities by preventing over-fitting, a problem that plagues all deep learning algorithms. As shown in Fig 2, the output of this layer represents the final CNN output.

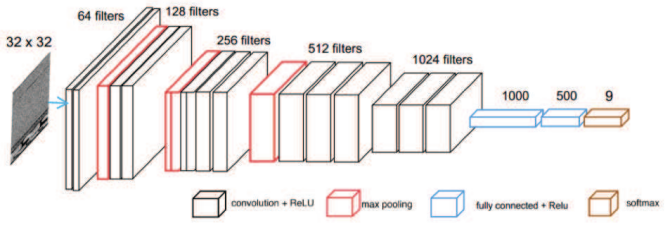

Fig. 2: Five layered CNN architecture [16].

The main parts of the layers shown in Fig. 2 can be explained as follow. The input layer will store the image's pixel values, as is typical of other ANN types. The convolutional layer will calculate the scalar product between the weights of the input volume-connected region and the neurons whose output is connected to local regions of the input. The rectified linear unit, also known as ReLu, aims to implement the output of the activation function, such as sigmoid. To further reduce the number of parameters in that activation, the pooling layer simply applies down sampling along the spatial dimension of the input. Finally, the fully-connected layers will carry out the same tasks as those of conventional ANNs and make an effort to derive class scores from the activations, which can be used for classification. Additionally, it is suggested that ReLu be applied in between these layers to enhance performance.

IV. Methodology and Dataset

In this study, we classify malware families using the CNN model by treating malware as RGB images. An open image database is the MaleVis (Malware Evaluation with Vision) dataset that contains 14226 from $(25 + 1)$ different malware categories RGB images. 9100 of them are for training and approximately 5126 RGB images for validation [14]. Malware classes include Adposhel, Agent, Allaple, Amonetize, Androm, Autorum, BrowseFox, Dinwod, Elex, Expiro, Fasong, HackKMS, Hlux, Injector, InstallCore, MultiPlug, Neroeklami, Nethta, Rugrun, Sality, Snarasite, Stantinko, VBA, VBKrypt and Vilsel. About 500 samples are chosen from each class. They are grouped into malware, trojans, worms, and viruses. Each class has 350 worth of samples for testing and different samples for training. The images must be the same size (224×224) in order to be fed into a CNN model. The dataset families are demonstrated in TableI. CNN is a type of deep learning model for processing such things as images. It can be modeled by complex mathematical expressions, which are generally divided into multiple layers as described in [17].

Adam Optimizer Adaptive Moment Estimation (AOAME) is adopted in this study. It is an optimization technique algorithm. When dealing with complex problems with numerous features, the approach is incredibly effective. It is effective and uses less memory. It combines the "gradient descent with momentum" algorithm and the "RMSP algorithm" intuitively [19].

TABLE I: Description of MaleVis Dataset. (Alphabet order)

Code	Malware	Sapmles	Malware Type
1	Adposhel	494	Adware
2	Agent	470	Trojan
3	Allaple	478	Worm
4	Amonetize	497	Adware
5	Androm	500	Win32
6	Autorum	496	Worm
7	BrowseFox	493	Adware
8	Dinwod	499	Trojan
9	Elex	500	Adware
10	Expiro	501	Virus
11	Fasong	501	Worm
12	HackKMS	499	Worm
13	Hlux	500	Trojan
14	Injector	495	Virus
15	InstallCore	500	Adware
16	MultiPlug	499	Adware
17	Neroeklami	500	-
18	Nethta	497	Virus
20	Rugrun	485	Virus
21	Sality	499	Virus
22	Snarasite	500	Trojan
23	Stantinko	500	Trojan
24	VBA	500	Virus
25	VBKrypt	496	Trojan
26	Vilsel	496	Trojan
19	Others[a]	1832	-
	Total	14226	

[a] This item has label 19 in Fig. 1.

Fig. 3 demonstrates sample color-based images (RBG) for all MaleVis dataset family used in the classification model.

V. Experiment Results

Confusion matrix as a measure of classification model performance is used in this paper. It is a widely used metric for addressing classification issues. Both binary classification and multi-class classification issues can be solved using it. The rows of the confusion matrix in Fig. 4 represent the actual

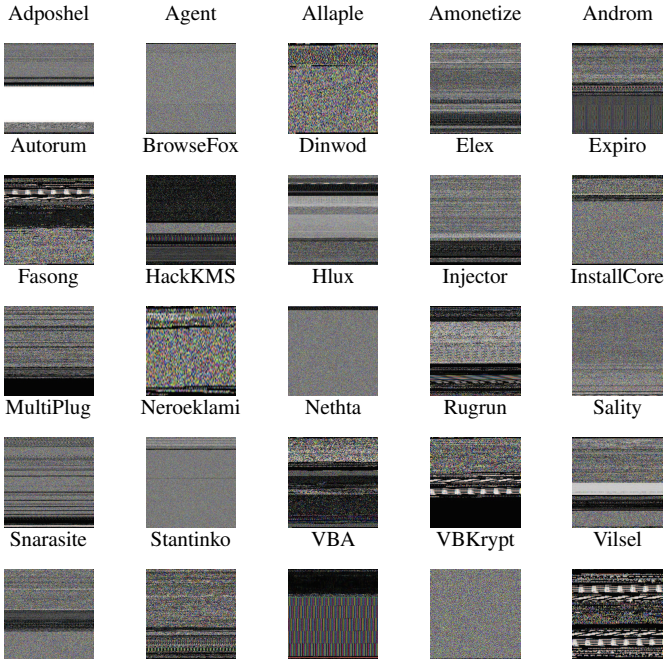

Fig. 3: Samples from MaleVis dataset family listed in Table I .

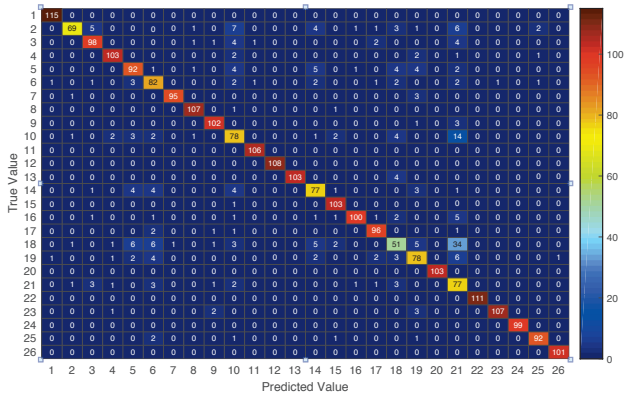

Fig. 4: Confusion matrix for the CNN implemented on the MaleVis dataset.

classes the outcomes should have been, while its columns represent the predictions values the model has made.

We employ the Tensorflow library to put our framework into practice. A useful Python package that makes it easier to gather data, train models, and provide predictions. The training dataset is initially randomly divided into 30% for validation and 70% for training. As stated, we ran the CNN model on the MaleVis dataset. Since the Malimg dataset is RGB images, in order to speed up training and decrease overfitting, we used the input layer shape $(3, 64, 64)$. With dropout rates of 0.25 and 0.50 each, we added two additional layers. The Adam optimizer described in Section IV a sigmoid nonlinear activation function, and a categorical-cross entropy loss function were all used by the CNN classifier. ReLU is also employed in convolutional and fully connected layers as an activation function. We put CNN, our envisioned network, through 30 training cycles. Fig. 4 provides the confusion matrix for multiclass malware prediction using the CNN model. Our model delivered solid results, with very good classification accuracy.

VI. Conclusion and Future Work

In this paper, we introduce a malware classification method using the CNN model to solve malware recognition problems without needing a handmade feature design or further engineering techniques. We have successfully trained the CNN model using the MaleVis dataset to automatically classify the data into 26 different malware families. Due to the effectiveness and efficiency of CNN in detecting malware images, the model's classification speed was faster than that achieved by other models that use traditional techniques [20], [21]. Future work will be focused on obtaining results using additional models for image classification. We also prepare all other malware byte files and convert them into color RGB images before they are sent to the classification department.

References

[1] CISCO Technical Team, "What's Malware?", https://www.cisco.com/c/en/us/products/ security/advanced-malware-protection/what-is-malware, January, 2023.

[2] H. Lallie, L. Shepherd, J. Nurse, A. Erola, G. Epiphaniou, C. Maple and X. Bellekens, "Cyber Security in the Age of COVID-19: A Timeline and Analysis of Cyber-crime and Cyber-attacks During the Pandemic," in Computers & Security, vol 105, pp 1-20, 2021.

[3] Sonic Wall Cyber Threat Report, https://www.sonicwall.com/medialibrary/en/white-paper/mid-year-2022-cyber-threat-report.pdf, Mid year, 2022.

[4] Annual Number of Malware Attacks Worldwide from 2015 to 2022, https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/," June, 2022.

[5] Kaspersky Cybercrime, Inc. "How profitable is the business?," "https://blog.kaspersky.com/cybercrime-inc-how-profitable-is-thebusiness/," December, 2014.

[6] A. Singh, A. Handa and S. Shukla, "Malware Classification Using Image Representation, in Third International Symposium on Cyber Security, Cryptology and Machine Learning (CSCML 2019), June, 2019.

[7] S. Akarsh, K. Simran, P. Poornachandran, V. K. Menon and K. P. Soman, "Deep Learning Framework and Visualization for Malware Classification," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 1059-1063, doi: 10.1109/ICACCS.2019.8728471.

[8] E. Gandotra, D. Bansal and S. Sofat, "Malware Analysis and Classification: A Survey," Journal of Information Security, pp. 56- 64, April, 2014.

[9] M. Alom, T. Taha, C. Yakopcic, S. Nasrin , M. Hasan and B. Essen, "A State-of-the-Art Survey on Deep Learning Theory and Architectures, in Electronics (MDPI), pp 1-67, March, 2019, DOI: 10.3390/electronics8030292.

[10] 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 Mining (DMIN), in Twelfth International Conference on Data Mining , July, 2016, USA.

[11] T. Bounouh, Z. Brahimi, A. Al-Nemrat and C. Benza,"A Scalable Malware Classification based on Integrated Static and Dynamic Features," Part of the Communications in Computer and Information Science book series (CCIS), volume 630, January, 2017.

[12] T. Isohara, K. Takemori and A. Kubota, "Kernel-based Behavior Analysis for Android Malware Detection," 2011 Seventh International Conference on Computational Intelligence and Security, Sanya, China, 2011, pp. 1011-1015, doi: 10.1109/CIS.2011.226.

[13] S. Ni, Q. Qian and R. Zhang, "Malware Identification Using Visualization Images and Deep Learning," Computers & Security, vol. 77, pp.871-885, 2018.
[14] "MaleVis: A Dataset for Vision Based Malware Recognition," https://web.cs.hacettepe.edu.tr/ selman/malevis/index.html, 2019.
[15] R. Yamashita, M. Nishio, R. Kinh Do and K. Togashi, "Convolutional Neural Networks: an Overview and Application in Radiology, European Society of Radiology Journal (ESR), pp. 611-629, 2018, doi: 10.1007/s13244-018-0639-9.
[16] Convolutional neural network (CNN, or ConvNet), https: //cs231n.github.io/convolutional-networks/, 2019.
[17] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.
[18] U. Tayyab, F. Khan, M. Durad, A. Khan and Y. Lee, " A Survey of the Recent Trends in Deep Learning Based Malware Detection," in Electronics (MDPI), Journal of Cybersecurity and Privacy, pp. 800-829, February, 2022, doi:10.3390/jcp2040041.
[19] D. Kingma and J. Ba,"Adam: A Method for Stochastic Optimization," 3^{rd} International Conference for Learning Representations (ICLR), San Diego, USA, 2015
[20] S. Patil, V. Varadarajan, D. Walimbe, S. Gulechha, S. Shenoy, A. Raina, and K. Kotecha, "Improving the Robustness of AI-Based Malware Detection Using Adversarial Machine Learning Algorithms," in Multidisciplinary Digital Publishing Institute (MDPI) December 2022.
[21] A. Bozkir, A. Cankaya and M. Aydos "Utilization and Comparison of Convolutional Neural Networks in Malware Recognition" Proceedings of the 27^{th} Signal Processing and Communications Applications Conference (SIU); Sivas, Turkey, April, 2019.