A Deep Learning-Based Approach for Malware Classification Using Machine Code to Image Conversion

S. Yaseen[1], M. M. Aslam[2], M. Farhan[3], M. R. Naeem[4], A. Raza[5]

[1,2,3,4,5]*Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus*

[3]farhansajid@gmail.com

Abstract- Malware presents a formidable threat to computer security, with some antivirus companies reporting over five million malware samples per day. Due to the sheer volume of malware, security teams cannot feasibly address each instance individually, necessitating the use of malware classification schemes. The size, type, and complexity of malware continue to grow, as hackers and attackers often create systems that can automatically transfer and encrypt code to evade detection. Classic machine learning techniques that rely on handmade feature vectors are not effective for classifying malware families. In contrast, deep convolutional neural networks have demonstrated efficiency in detecting and classifying malware. This article proposes a new system for classifying malware into families by transforming malware binaries into grayscale images and applying convolutional neural networks. Our approach provides a new method for multi-class classification challenges in the field of cybersecurity, and it outperforms traditional machine learning techniques. By utilizing deep learning models to classify malware, we can enhance our ability to detect and mitigate threats in real-time. Our work emphasizes the potential of advanced machine learning techniques in cybersecurity and calls for further research in this area.

Keywords- Malware classification, Deep learning CNN, bytes code, Machine code, Image conversion, Cybersecurity

I. INTRODUCTION

In recent years, the number and complexity of malware attacks have increased dramatically, presenting a significant challenge to computer security teams. With some antivirus companies reporting over five million malware samples per day, it is impractical to address each instance individually. Instead, the use of malware classification schemes has become a critical approach for detecting and mitigating threats.

Classifying malware based on its family is an important task in cybersecurity, as it allows security teams to develop targeted defenses against specific types of malware. However, traditional machine learning techniques that rely on handmade feature vectors are not effective for classifying malware families, as malware samples are highly variable and frequently encrypted or obfuscated. As a result, recent research has focused on the use of deep learning models for malware classification.

In particular, deep convolutional neural networks (CNNs) have demonstrated high efficiency and accuracy in detecting and classifying malware based on features extracted from binary code. In this paper, we propose a new system for classifying malware families based on grayscale images generated from machine code. Our approach employs deep CNNs to extract features from these images, providing a new method for multi-class classification challenges in the field of cybersecurity.

The main contributions of our work include the development of a new methodology for malware classification using deep learning models and the use of machine code to image conversion to generate grayscale images for classification. We evaluate the effectiveness of our approach using a real-world dataset and compare it with traditional machine learning techniques. Our results show that our approach outperforms traditional methods and provides an effective solution for malware family classification.

Malware attacks are becoming a most crucial threat to internet and network protection. According to a Symantec report, 123 million devices regularly record thousands of malicious threat actions per second [1]. They are top-ranked computer security threats. The number is staggering, as some anti-virus companies report over 5 million malware samples per day. Internet-based devices are connected worldwide and are anticipated to reach 200 billion by 2020 [2]. It can be complicated and complex, but it can be more accurate. The security team cannot deal with all this malware simultaneously. A malware classification scheme is typically required to prioritize these incidents.

The malware attacks are growing day by day in mobile devices and the Internet of Things (IoT). The Complex Systems software environment and sensory devices make it easier for opponents to launch attacks on the system. Malware is malicious software that affects digital systems' functionality, privacy, and consistency—different types or families of this malware include Trojan horses, backdoors, and worms. Malware is a significant security risk to systems these days [3].

Malware writers use a collection of methods and techniques to write a program that hides their identities. Therefore, discovering malware families or malware types is the most complicated part. With traditional anti-malware, it is very complex to deal with the enormous amount of malware that is increasing every day. Computer scientists and anti-virus companies have initiated using machine learning models to solve this problem. Academic researchers and developers have proposed many machine learning classifiers, including Neural networks and logistic regression for classifying malware [4]. The first step towards efficiently classifying and estimating many such files is clustering and categorizing their families. Moreover, such clustering criteria might be helpful to new files found on the system can help to identify them as malicious and from a particular family. To investigate this domain, we need malware files, and their families should be grouped and used to identify new malware based on these clusters. Currently, malware manufacturing has become a vast and well-established market. In the internet era, several malware attacks occur that cause significant security threats to commercial organizations and regular clients in our country. It is necessary to improve malware categorization methods to deal with this malware's rapid development that allows variations of malware files that fit a similar family. Therefore, the vision of developing technologies that properly categorize malware according to that one family, regardless of malware type, seems to be an advantageous and helpful way to deal with the rapid development of malware. Here are some areas where a solution to this problem can be beneficial. One is the anti-malware generator, and the other is the identification of the malware developer.

Malware classification is beneficial for predictors because it'll help them better understand the malware behavior. Malware having similar structures is categorized into one cluster. Similarly, we know the malware family to which they belong. We have a general idea about its behavior. Malware analysis and classification is a rapidly growing domain and demands high awareness because of high-tech development evolutions in the Digital software environment, mobile environment, social networks, smart cities, cloud computing, and the Internet of Things (IoT). Researchers have recently achieved excellent results by using recurrent neural networks (RNNs) for speech recognition and handwriting recognition [5]. Various scientists have used machine learning techniques to identify and classify malware. Detection of malware is based on two stages. The first is extracting features from images, and the second is malware classification[6]. Using theoretical techniques to detect malware takes too much time and gives unacceptable results. The latest system learned from this manual survey is implemented with automated software that improves performance. The famous quote from Fred R. Bernard, “Pictures deserve a million of words,” encouraged us. In this article, we evaluate whether the problem also applies to malware Visualizations have permanently been established to help find a detailed picture of some framework or records. Logically, it can make more visual sense than any other expression [7]. The question here is whether these sequential models can also provide high accuracy in detecting malware families? Therefore, we used a deep learning model before classifying different malware families in this thesis. Extremely malware classification techniques are created on feature vectors representing malware traits. This study categorizes existing and new malware that is being developed. Anti-malware companies can quickly build anti-malware for existing and unknown malware. The overall contributions to this study are: Use the DL model to classify different types of malwares. We have identified work related to the image-based classification of malware. Our model does not require feature engineering or domain expertise, such as binary disassembly, reverse engineering, or assembly language. Our method will use real-time classification because the preprocessing time is short. After all, we approved the image dataset directly as input to the model. We compared our model against the Microsoft dataset containing nine different malware families to classify new malware, developing an advanced system by using single or hybrid deep learning models used for the malware classification that is computationally cost-effective, scalable, and efficient to use.

II. Problem Statement

Malware is common in this digital world. These types of software are developed for various purposes, e.g., the steal data, corrupting executable files, and extracting information from different computers networks. Anti-virus and anti-malware can detect these types of software. There is a need to identify such a type of software, which can be categorized and identified efficiently by using a deep learning approach. Automatic categorization can help the anti-virus or anti-malware software block such software. This type of work has been done in the literature, but we improved the accuracy and efficiency of the algorithms.

III. Limitations

Despite the promising results, there are a few limitations to our proposed approach. One limitation is the need for a significant amount of computational power for training deep learning models, which can be costly and time-consuming. Additionally, the dataset used in our experiments may not be fully representative of the real-world malware landscape, which could affect the generalizability of our approach. Another limitation is that the proposed system currently only works for malware classification and does not provide a solution for malware detection.

To overcome the limitations of our proposed approach, future research could focus on developing more efficient deep learning models that require less computational power for training. Another direction for future research could be to expand the dataset used in our experiments to include a more diverse range of malware samples, which would increase the generalizability of our approach. In addition, future work could focus on combining our proposed approach with other techniques for malware detection to create a more comprehensive solution for malware analysis. Finally, it would also be interesting to investigate the use of other types of machine learning models and feature extraction techniques to further improve the accuracy of malware classification and detection.

The remainder of this paper is structured as follows: Section 2 describes related work in malware classification and deep learning. Section 3 presents the methodology used in our study, including details of the dataset, the image generation process, and the CNN architecture. Section 4 presents our experimental results and compares our approach with traditional machine learning techniques. Section 5 discusses the limitations of our study and identifies potential areas for future research. Finally, Section 6 provides a summary of our contributions and highlights the potential of deep learning models in cybersecurity.

IV. Related Work

For creative malware classification, many researchers have performed malware visualization to achieve extreme accuracy in a short time. This segment of the synopsis presents related works concerning malware recognition, visualization, and classification based on some machine learning models and deep learning models.

K. Omar [8]. We have proposed a scheme malware-detection built on the various behaviors of malware executables. The main thing is to identify similarities in the conduct of malware samples. In this technique, the researchers implement malware executables in a virtual environment. When they implement malware, they produce things like images and patterns. Use a color map to check the order of the designs and some statistical techniques to prevent the behavioral similarity of these images. They achieved higher accuracy of 95.91% to 98% by taking 1,102 malware image samples and taking them from 12 different malware families. However, this approach of exceptional malware in a virtual environment proves to be very time-consuming [9].

The authors have first introduced a malware classifier that worked on feature extraction and then used a support vector machine to classify the malware. Using this technique, we achieved a classification accuracy of 97.4 using a dataset containing data from 25 malware families with 9339 sample files. However, as a result, conventional techniques require very long computational times for feature analysis. To solve this problem, we use deep learning for image classification. New technologies are being proposed related to the proposed method. This method uses CNN to classify malware [5]. They applied various architectures to its base model, but it is relatively shallow in that environment.

J-Sin Luo proposed a method to run on GPU using Tensor-Flow, which has a short processing time. However, this scheme does not work on virtual devices, and it is not possible to determine the behavior of the malware. This technology is based on image recognition. However, this method still has some flaws. If the developer rewrites the code differently, the result will change, and this technique will fail [10]. Ayan Dey et al. require data on already existing methods of image processing-based malware detection. They used an entropy filter to find patterns in the image and got better results [11] than J. Chen [12]. R. Zhang et al., many techniques suggest that opcode sequences taken from malware executables files are converted into a visual matrix displayed in RGB color pixels and converted into an entropy graph to show similarities between the images. However, these techniques only worked for Windows PE files, not packed data samples [13]. Here, a new feature fusion technique was proposed to reconstruct features extracted from pre-trained Alex Net. They used 25 classes to recognize the malware and used the images extracted to classify it using various SVM, Decision Tree, and K-nearest neighbor variants [14]. Using these Machine learning variants, they achieved 99.3 percent accuracy [15-16].

J. Chen [12] Introduced the latest system for malware detection, which uses feature extraction. They converted malware binary executable into a 2D gray-scale image. They generated a dataset of images and used visible patterns or characteristics to detect malware. Results had indicated that it was more accurate and significantly less time-consuming. To enhance the security of existing infrastructure, academic researchers look at malware examples to learn how they work and explore the strategies used by malware developers. Malware analysis is being used for the classification of malware. The process of determining which class a piece of malware belongs to is known as classification. Now that you have decided that the file is malware, you need to identify its family. There are two types of malware analysis: static and dynamic. Static evaluation is a method of analyzing without running a sample. At the same time, dynamic analysis is the procedure of running a sample and determining its behavior, like how it performed in different environments. But we used various approaches; we completed the analysis of malware files by converting executable malware files into Gray Scale images. They did not damage the system and did not require the file to be executed.

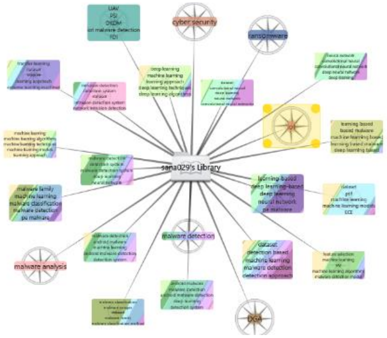


Figure 1: A graphical representation of the related works in the field of malware classification using deep learning techniques

Static Malware Analysis

Static malware analysis is a widely used approach for evaluating malware without running its binary file. This method is particularly useful for analyzing a variety of executable expressions, including those that are designed to avoid detection by antivirus software or other security measures. One of the advantages of static analysis is that it can be performed even when the actual code is not available. In such cases, analysts can evaluate the raw hex code or first decompose the binary file and then describe the assembler code.

There are several techniques that can be used for static analysis, including examining printable strings in a program, looking for file headers, exploding a program, and searching byte sequences. For example, by examining the strings present in a malware file, analysts can often determine what the malware is designed to do, how it communicates with other systems, and what kind of data it is targeting. Similarly, looking for file headers can reveal information about the format and structure of the file, as well as any dependencies it may have on other files or libraries.

Another common technique for static analysis is to explode a program into its constituent components, such as its code, data, and resources. This can help analysts gain a better understanding of how the program works and what its intended purpose may be. Searching for byte sequences can also be a useful approach for identifying patterns or signatures that are characteristic of specific types of malware.

Despite its many advantages, static malware analysis also has some limitations. For example, it may not be able to detect malware that is designed to evade static analysis techniques or that is polymorphic in nature. In such cases, dynamic analysis or other techniques may be required to fully evaluate the malware. Additionally, static analysis can be time-consuming and resource-intensive, particularly when dealing with large or complex programs.

Overall, static malware analysis is an important technique for evaluating malware and detecting threats to computer security. By using a variety of approaches, such as examining printable strings, looking for file headers, and exploding a program, analysts can gain valuable insights into the structure, behavior, and purpose of malware, and develop effective strategies for mitigating the risks it poses.

Dynamic Malware Analysis

Dynamic analysis is another commonly used approach for evaluating malware, which involves running an executable file and analyzing its behavior to determine if it is malicious. This approach is particularly useful for detecting malware that is designed to evade static analysis techniques, such as polymorphic or metamorphic malware. Dynamic analysis typically involves running the malware in a virtualized environment or virtual machine (VM) that simulates the target system's operating environment while limiting the impact of the malware. One of the advantages of using a VM for dynamic analysis is that it allows for the creation of a snapshot of the system's state before the malware runs. This snapshot can then be used to quickly revert the system to its original state after the analysis is complete, reducing the risk of damage to the system. In addition to traditional dynamic analysis techniques, recent research has explored the use of image-based analytics for analyzing malware behavior. One such study employed image-based analytics to describe the timing of malware attacks, such as spear-phishing attacks. The study used color-coded images to represent the types of system connections that were successful during an attack, providing a visual representation of the attack pattern.

Despite its many advantages, dynamic analysis also has some limitations. For example, it may not be able to detect malware that is designed to detect and evade virtualized environments. Additionally, dynamic analysis can be time-consuming and resource-intensive, particularly when dealing with complex or advanced malware. Overall, dynamic analysis is an important approach for detecting and mitigating malware threats, particularly those that are designed to evade traditional static analysis techniques. By using a virtualized environment or VM, researchers can analyze malware behavior without risking damage to the host system. Furthermore, recent advances in image-based analytics are providing new opportunities for analyzing and understanding malware behavior [17].

Applying a single feature for malware sensing or categorization may not be sufficient to effectively detect or classify malware in real-world contexts. This is because malware authors often use obfuscation methods that can obscure the features used by the machine learning model. As a result, it is necessary to develop algorithms that can handle multiple features. Current techniques for malware classification can be broadly categorized into two types: built sets and data-level fusion. The built sets approach involves grouping features together into pre-defined sets, while the data-level fusion approach combines multiple databases into a single feature vector, which is then input into a machine learning algorithm.

One significant advantage of using convolutional neural networks (CNNs) for malware classification is that they can learn n-gram-like signatures without requiring a manually created list of many n-grams during training, as is the case with the n-gram-based approach. Instead, CNNs use convolutional layers to automatically identify n-gram-like patterns in the data, which can then be used for classification. This approach has several important implications, including the removal of traditional pipelines consisting of feature mining, selection, reduction, and classification. Instead, all of these steps are optimized together during the training of the CNN. By using CNN-based methods, it is possible to develop highly effective algorithms for malware classification that can handle multiple features and automatically learn n-gram-like patterns in the data. This can significantly improve the accuracy and efficiency of malware detection and classification in real-world contexts [18].

V. RESEARCH METHODOLOGY

This section presents a novel methodology for malware categorization using a deep learning system. Our approach introduces a new technique for the multi-class categorization problem by converting malware executables into grayscale images, which are then used to classify and identify the malware family. By identifying the malware family, analysts can gain insights into the behavior and type of malware, which can aid in the development of anti-malware solutions. The task of developing a comprehensive malware classification system that can handle a vast number of malicious codes and identify their families is challenging. In this section, we describe the steps taken for this classification using the methodology shown in Figure 1:

1. Data Collection and Preprocessing: The first step involves collecting a dataset of malware samples and preprocessing the data to prepare it for analysis. This includes removing irrelevant data, converting the executable files into binary code, and splitting the dataset into training and testing sets.
2. Visualization of Binary to Grayscale Image: The next step involves converting the binary code into grayscale images, which can be used as input data for the deep learning model. We use a simple transformation algorithm to generate grayscale images that accurately capture the characteristics of the binary code.
3. Model Training: Once the grayscale images have been generated, we use a deep convolutional neural network (CNN) to train a model for malware classification. The CNN is trained on the training set using backpropagation and stochastic gradient descent, optimizing for high accuracy and low loss.
4. Model Testing: The final step involves testing the trained model on the testing set to evaluate its performance. We use various evaluation metrics such as accuracy, precision, recall, and F1 score to assess the effectiveness of our approach.

Our proposed methodology offers several advantages over traditional machine learning techniques for malware classification. By converting malware executables to grayscale images, we can capture more detailed information about the code, which can improve the accuracy of the classification. Furthermore, by using a deep CNN for classification, we can automatically learn important features and patterns in the data, without the need for manual feature engineering. Overall, our methodology provides a promising approach for malware classification using deep learning models and image-based analysis. Future work in this area could focus on optimizing the CNN architecture and exploring the use of additional image transformation techniques to further improve the accuracy and efficiency of malware classification.

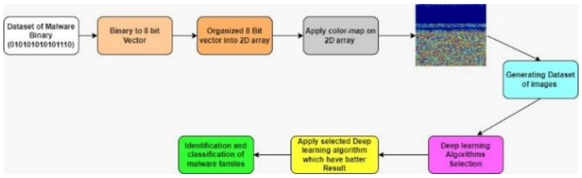

Dataset of Malware Binary (010101010101110)
Binary to 8 bit Vector
Organized 8 Bit vector into 2D array
Apply color-map on 2D array
Generating Dataset of images
Deep learning Algorithms Selection
Apply selected Deep learning algorithm which have batter Result
Identification and classification of malware familes

Figure 2. Methodology for malware categorization using deep learning system

BIG 2015, offered by Microsoft on the Kaggle platform, was used for technical evaluation of the proposed DLMD. These data have become academic references, with more than 50 articles citing them [19] The dataset is 0.5 terabytes in size, with 10868 training samples and 10868 testing samples. Because labels haven't been provided based on a test sample in the past, this study relied solely on trained data. There are nine malware families in the dataset. On the other hand, the frequency distributions of different families are plotted, while the frequency distributions of different families are plotted. This figure shows a significant imbalance in the classes of malware datasets. The most affluent type is Kelihos_ver3, with about 5000 samples, while the Simda class is the tiniest sample with about 40 samples. A description of the various malware families is given as in Figure 3.

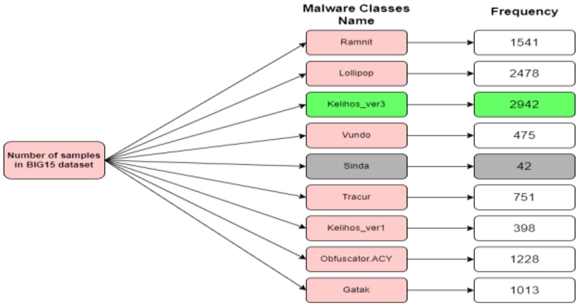


Figure 3. Overview of the BIG 2015 dataset used for technical evaluation

Several types of malware exist that pose significant threats to computer systems and user privacy. Some of the most notable examples include:

Lollipop: This malware displays advertisements in the browser, allowing a hacker to track user activity and potentially steal sensitive information.

Simda: A type of malware that steals credentials from users' computers and opens a backdoor for hackers, allowing them to access the system remotely.

Kelihos ver1 and Kelihos ver3: These trojans can take complete control of a user's computer and spread by sending spam emails from that computer to other computers, potentially infecting entire networks.

Ramnit: This malware is designed to steal user credentials such as passwords and credit card numbers, as well as shut down security software to avoid detection.

Vundo: This malware is often responsible for the installation of pop-up advertising and other dangerous software, which can compromise user privacy and security.

Obfuscator: This is a tool that can be used to obfuscate information, making it harder for analysts and security software to detect and identify malware.

ACY: These are obfuscated malware, and their objectives may be similar to those of the malwares listed above.

Traceur: This malware is designed to generate fraudulent reviews on search engines, allowing the author to earn money through deception.

Gatak: A Trojan horse that appears to be legitimate but infects computers with harmful programs.

The BIG2015 training dataset includes two files for each malware: a byte file and an assembler code (.asm file). These files can be used to train machine learning models for malware detection and classification. By understanding the characteristics of different types of malware and developing effective detection and mitigation strategies, we can better protect computer systems and user privacy from the threats posed by these malicious programs.

ASM File

The ASM file is a type of malware assembly code that contains information about function calls and variable assignments. A sample of the ASM file is shown in Figure 3. The main section of the ASM file typically consists of a bss section and a text section, although there may be other sections that contain information about secondary features. The number and type of sections present in an ASM file can vary depending on the type of malware, with obfuscated viruses typically having different sections than Ramnit malware, for example. One of the primary sections in an ASM file is the .text section, which contains the malware's actual code. This section typically includes instructions for the malware to carry out specific tasks, such as stealing user credentials or infecting other systems. Other sections in the ASM file may contain information about data variables, function parameters, and other important features of the malware. By analyzing the ASM file, researchers can gain a deeper understanding of the behavior and characteristics of different types of malware, which can aid in the development of more effective detection and mitigation strategies. Additionally, machine learning models can be trained on ASM files to identify patterns and features associated with different types of malware, enabling automated detection and classification of these malicious programs. Here are some of the primary sections and their purposes:

•.text: This section contains the malware's actual code.

•.data: This section holds variables or data that have been initialized.

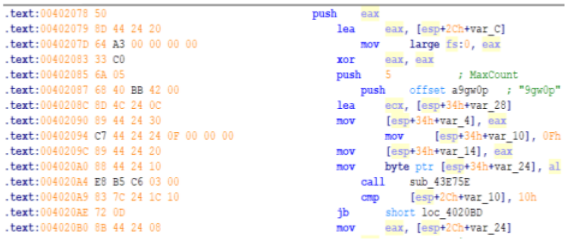
•.rdata: This is where you put read-only data or constants that shouldn't be altered while the application is executing.

•.bss: This section contains variables that have not been initialized and can be used during execution.

•.idata: This file includes information about the folders and programmer that your code imports while running.

•.edata: This file contains information on the data that the virus outputs while it is active.

•.rsrc: This file includes the program's actual resources.



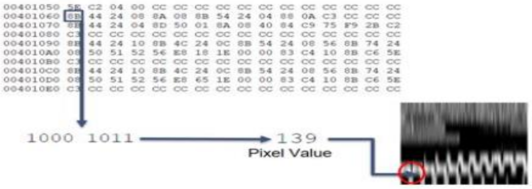
```
.text:00402078 50                          push    eax
.text:00402079 8D 44 24 20                     lea     eax, [esp+2Ch+var_C]
.text:0040207D 64 A3 00 00 00 00                   mov     large fs:0, eax
.text:00402083 33 C0                           xor     eax, eax
.text:00402085 6A 05                           push    5               ; MaxCount
.text:00402087 68 40 BB 42 00                      push    offset a9gw0p   ; "9gw0p"
.text:0040208C 8D 4C 24 0C                     lea     ecx, [esp+34h+var_28]
.text:00402090 89 44 24 30                     mov     [esp+34h+var_4], eax
.text:00402094 C7 44 24 24 0F 00 00 00                 mov     [esp+34h+var_10], 0Fh
.text:0040209C 89 44 24 20                     mov     [esp+34h+var_14], eax
.text:004020A0 88 44 24 10                     mov     byte ptr [esp+34h+var_24], al
.text:004020A4 E8 B5 C6 03 00                      call    sub_43E75E
.text:004020A9 83 7C 24 1C 10                      cmp     [esp+2Ch+var_10], 10h
.text:004020AE 72 0D                           jb      short loc_4020BD
.text:004020B0 8B 44 24 08                     mov     eax, [esp+2Ch+var_24]
```

Figure 4. A snippet of the assembly code from an ASM file, which is a hexadecimal representation of a malware portable executable (PE) file

Byte File

Byte files are representations of malware portable executable (PE) files in hexadecimal format. Each line in the byte file is called a record and consists of two parts. The first part is the offset of the instruction's memory address, while the second part is the statement itself, represented by a hexadecimal pair. These byte files can be generated from portable executables using tools such as the IDA disassembler. The byte files directly map the assembler instructions that exist in the ASM file. To utilize these files for classification using a deep learning system, we convert them into images. Each hexadecimal pair in the byte file is treated as a single decimal number, which is used as the pixel value for the corresponding image. The resulting image is then resized to a standard size of 32x32 pixels, as shown in Figure 5.

By converting the byte files to images, we can leverage the power of image-based analysis and deep learning models for malware classification. The resulting images capture important features of the malware code, which can aid in the accurate classification of different types of malware. Overall, our methodology of converting byte files to images provides a promising approach for detecting and mitigating the threat posed by malware.



```
00401050 5E C2 04 00 CC CC CC CC CC CC CC CC CC CC CC CC
00401060 8B 44 24 08 8A 08 8B 54 24 04 88 0A C3 CC CC CC
00401070 8B 44 24 04 8D 50 01 8A 08 40 84 C9 75 F9 2B C2
00401080 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
00401090 8B 44 24 10 8B 4C 24 0C 8B 54 24 08 56 8B 74 24
004010A0 08 50 51 52 56 E8 18 1E 00 00 83 C4 10 8B C6 5E
004010B0 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
004010C0 8B 44 24 10 8B 4C 24 0C 8B 54 24 08 56 8B 74 24
004010D0 08 50 51 52 56 E8 65 1E 00 00 83 C4 10 8B C6 5E
004010E0 C3 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC
```

Figure 5. Conversion of Byte File to Image for Malware Classification

Convert Binary Files to 8-bit Vector

In this case, CNN searches for the most relevant features in the image of a particular malware family for classification [20]. A technique that turns the binary PE file into a sequence of 8-bit vectors or hexadecimal values can be used to convert malicious binary files into pictures. In the range 00000000 (0) to 11111111, an 8-bit vector can be expressed (255). Figure 5 shows how each 8-bit vector represents a number and can be transformed to pixels in a virus image.

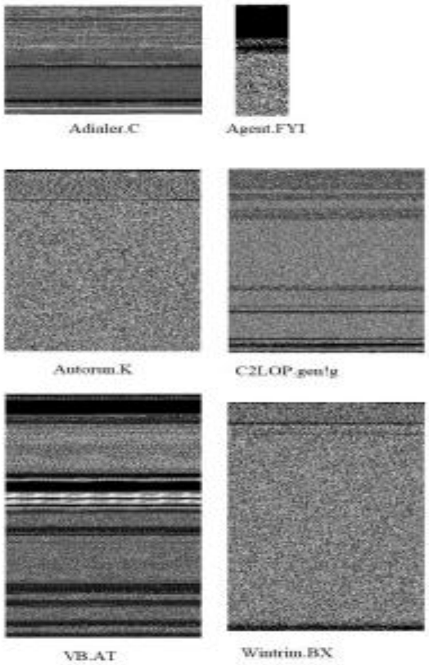


Figure 6: Sample images of malware belonging to different families. Each grayscale image was generated by converting the malware binary to an 8-bit vector and then treating each vector as a pixel

Convert 8-bit Vector to a Grayscale Image

The process of converting binary files to images begins by first converting the binary file to 8-bit vectors. These vectors are then used to create grayscale images, with each vector representing a pixel in the image and the intensity of the pixel being determined by the value of the corresponding vector element. This process is illustrated in Figure 6, which shows the conversion of a malware binary to a grayscale image. By using grayscale images instead of the raw binary files, we can leverage the power of image-based analysis and deep learning models for malware classification. These images capture important features and patterns in the binary code, which can aid in the accurate identification and classification of different types of malware. Overall, our proposed approach of converting binary files to grayscale images provides a promising method for detecting and mitigating the threat posed by malware. By utilizing deep learning models trained on these images, we can better understand the behavior and characteristics of different types of malware, enabling us to develop more effective strategies for protecting computer systems and user privacy.

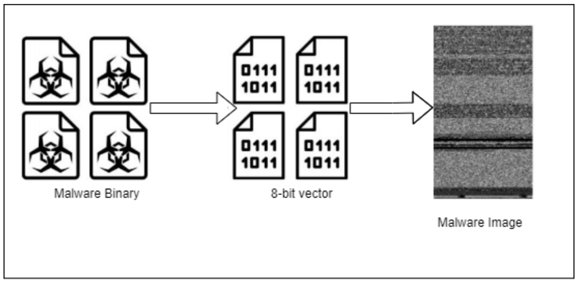


Figure 7: Process of creating a grayscale image from binary of a malware program

The grayscale images created from binary files exhibit a visually distinguishable interclass relationship, meaning that images belonging to the same malware family have similar visual characteristics, while those belonging to different families have distinct differences. This visual relationship between images makes image-based classification a powerful and effective method for this dataset. As shown in Figure 7, a typical grayscale image of malware contains important features and patterns that can aid in accurate classification and identification of the malware family. By utilizing deep learning models trained on these images, we can classify different types of malware with high accuracy, enabling us to detect and mitigate the threat posed by malware effectively. Overall, our proposed methodology of converting binary files to grayscale images provides a promising approach for improving the effectiveness of malware classification and detection. By leveraging the power of image-based analysis and deep learning models, we can better understand the behavior and characteristics of different types of malware, enabling us to develop more effective strategies for protecting computer systems and user privacy.

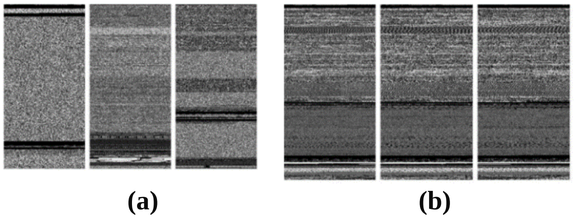


Figure 8. Grayscale images of malware: (a) unrelated classes of malware and (b) related classes from the same malware family [21].

Grayscale Images to RGB Conversion

In this section of our research paper, we explore the process of converting grayscale images to colored images. As discussed in the previous section, we converted binary files to grayscale images to leverage the power of image-based analysis and deep learning models for malware classification. To further improve the accuracy of our classification system, we applied a color map to the grayscale images, generating a dataset of 10,868 colorful images.

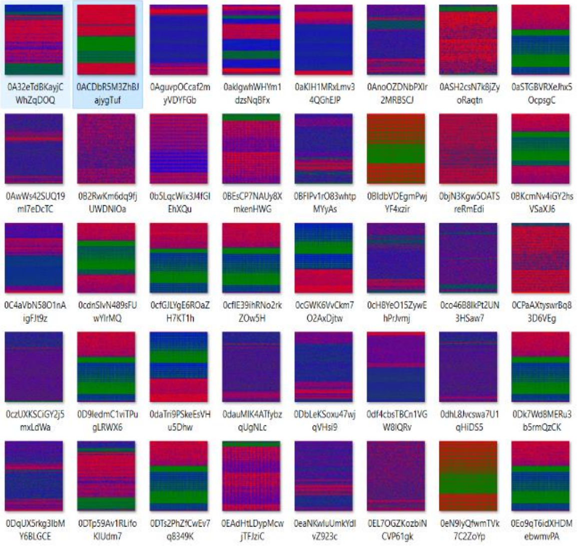


Figure 9: Colored malware images dataset resulting from the application of a color map to grayscale images, enabling the capture of more subtle features and patterns in the malware code, and improving the accuracy of the classification system.

This process involved mapping different colors to specific pixel values in the grayscale images, thereby creating a visually distinct representation of each malware family. By using colored images instead of grayscale images, we can capture even more subtle features and patterns in the malware code, enabling us to classify different types of malware with even greater accuracy. This approach represents a significant advancement in the field of malware classification and detection, providing a powerful and effective method for detecting and mitigating the threat posed by malware as shown in Figure 9.

VI. Results and Discussion

In this section of our research paper, we present the results of our proposed methodology for malware classification using deep learning models and image-based analysis. We trained our model using different epoch values, ranging from 1400 to 2000 epochs, to study the behavior of the model for different inputs. As shown in Figure 10, our operating system structure involved the installation of a malware classification image set and extracting a set of classifications. Our model achieved an overall accuracy of 82% after dividing the dataset into 70% for training and 30% for validation using our neural network. The model was trained on a dataset of 10,868 malware images and tested on 5,814 malware classifications, achieving an accuracy of 82% while classifying 9 different categories of malware. These results demonstrate the effectiveness of our proposed methodology for malware classification, highlighting the potential for using advanced machine learning techniques in cybersecurity. Our approach represents a significant step forward in the field of malware detection and mitigation, enabling us to better protect computer systems and user privacy from the threat posed by malware as shown in Figure 10.

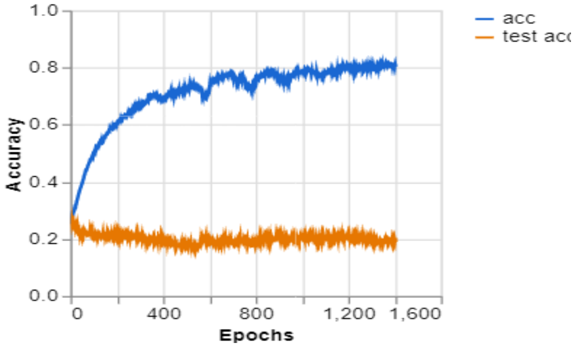


Figure 10: Malware classifications accuracy result

In this paper, we introduced a new algorithm called "Classify Malware Pictures" that effectively addresses the problem of malware classification with high accuracy, achieving an overall accuracy of 82%. This was accomplished through the use of convolutional neural networks (CNN), which allowed us to effectively extract important features and patterns from the malware images.

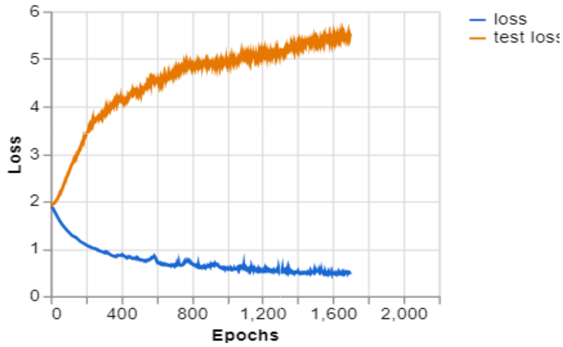


Figure 11: Malware classifications loss result

In addition, we also provided a dataset of 10,868 malware images, which can be used for further research and development in the field of cybersecurity. This dataset, coupled with our proposed methodology, represents a significant contribution to the field of malware detection and mitigation, providing a powerful tool for protecting computer systems and user privacy from the threat posed by malware. Overall, our approach represents a significant advancement in the field of malware classification, demonstrating the potential for using advanced machine learning techniques to effectively detect and mitigate the threat posed by malware. We hope that our work will inspire further research and development in this area, enabling us to develop even more effective strategies for protecting computer systems and user privacy from the ever-evolving threat of malware.

VII. Conclusion and Future Work

In this work, we presented a novel approach to malware classification using image-based analysis and deep learning techniques. Specifically, we represented malware binaries as images and used a pre-trained deep learning model for image recognition to classify the samples. Several experiments were conducted to evaluate the effectiveness of our approach, varying the datasets, classification levels, and learning methods. Our multi-class experiment was particularly impressive, achieving high accuracy across a wide range of malware families. Our approach has several strengths, including its robustness, which makes it well-suited for real-world applications. Moreover, there is still much room for future research in this area, and a variety of interesting solutions could be explored. For example, further studies could investigate the robustness of image-based techniques in more depth, or explore new ways to improve the accuracy and efficiency of our approach. Overall, we believe that our work represents an important step forward in the field of malware detection and classification, demonstrating the potential for using advanced machine learning techniques to more effectively combat the threat posed by malware. Specifically, our deep learning DEEP CNN has been shown to be an effective tool for mapping malware and achieving accurate classification, and we hope that our work will inspire further research in this area.

In terms of future work, there are several avenues for further research in the area of malware classification using image-based analysis and deep learning techniques. One promising direction for future work would be to scale up the dataset used in our study. Although our dataset of 10,868 malware images was significant, larger datasets could be collected and analyzed to improve the accuracy and generalizability of the approach. Additionally, more research is needed to quantify the robustness of image-based techniques in detecting new and emerging types of malware. While our approach achieved good accuracy, there is still room for improvement in terms of efficiency. Future work could focus on developing more efficient deep learning models or optimizing the training process to reduce the time and computational resources required for classification. Finally, transfer learning could be a promising avenue for future research in this area. By adapting pre-trained deep learning models to new tasks, it may be possible to improve the accuracy and efficiency of malware classification. Overall, continued development and exploration of image-based malware analysis using deep learning techniques could lead to significant advancements in the field of cybersecurity.

REFERENCES

[1] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture," *Computer Networks*, vol. 171, p. 107138, 2020.

[2] W. El-Shafai, I. Almomani, and A. AlKhayer, "Visualized malware multi-classification framework using fine-tuned CNN-based transfer learning models," *Applied Sciences*, vol. 11, no. 14, p. 6446, 2021.

[3] J. Su, D. V. Vasconcellos, S. Prasad, D. Sgandurra, Y. Feng, and K. Sakurai, "Lightweight classification of IoT malware based on image recognition," in *2018 IEEE 42Nd annual computer software and applications conference (COMPSAC)*, 2018, vol. 2, pp. 664-669: IEEE.

[4] R. Agrawal, J. W. Stokes, K. Selvaraj, and M. Marinescu, "Attention in recurrent neural networks for ransomware detection," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3222-3226: IEEE.

[5] M. Kalash, M. Rochan, N. Mohammed, N. D. 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)*, 2018, pp. 1-5: IEEE.

[6] L. Liu and B. Wang, "Malware classification using gray-scale images and ensemble learning," in *2016 3rd International Conference on Systems and Informatics (ICSAI)*, 2016, pp. 1018-1022: IEEE.

[7] A. Singh, A. Handa, N. Kumar, and S. Shukla, "Malware classification using image representation," *Cyber Security Cryptography and Machine Learning*, 2017.

[8] R. Sihwail, K. Omar, and K. A. Z. Ariffin, "A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 4-2, p. 1662, 2018.

[9] H. Naeem, B. Guo, and M. R. Naeem, "A light-weight malware static visual analysis for IoT infrastructure," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2018, pp. 240-244: IEEE.

[10] J.-S. Luo and D. C.-T. Lo, "Binary malware image classification using machine learning with local binary pattern," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 4664-4667: IEEE.

[11] A. Dey, S. Bhattacharya, and N. Chaki, "Byte label malware classification using image entropy," in *Advanced Computing and Systems for Security*: Springer, 2019, pp. 17-29.

[12] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-g. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187-3196, 2018.

[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] D. Xue, J. Li, T. Lv, W. Wu, and J. Wang, "Malware classification using probability scoring and machine learning," *IEEE Access*, vol. 7, pp. 91641-91656, 2019.

[15] M. Nisa *et al.*, "Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features," *Applied Sciences,* vol. 10, no. 14, p. 4966, 2020.

[16] L. Ghouti and M. Imam, "Malware classification using compact image features and multiclass support vector machines," *IET Information Security,* 2020.

[17] S. Venkatraman, M. Alazab, and R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection," *Journal of Information Security and Applications,* vol. 47, pp. 377-389, 2019.

[18] D. Gibert, C. Mateu, and J. Planes, "HYDRA: A multimodal deep learning framework for malware classification," *Computers & Security,* vol. 95, p. 101873, 2020.

[19] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security,* vol. 81, pp. 123-147, 2019.

[20] H. S. Anderson and P. Roth, "Ember: an open dataset for training static pe malware machine learning models," *arXiv preprint arXiv:1804.04637,* 2018.

[21] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge," *arXiv preprint arXiv:1802.10135,* 2018.