IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

COMPARISON OF DEEP LEARNING ALGORITHMS FOR MALWARE DETECTION

ROHINI N

STUDENT

SRI SARADA COLLEGE FOR WOMEN, TIRUNELVELI

ABSTRACT

Recent advances in computer technology enable the transfer of human existence from actual settings to virtual ones. This development has been hastened by Covid-19 illness. The focus of cybercriminals has also changed from physical to virtual life. This is so because committing a crime online is simpler than it is in real life. Unwanted software known as malicious software (malware) is regularly used by online criminals to launch cyberattacks. Advanced packing and obfuscation methods are being used by malware strains to continue their evolution. These obfuscation methods make malware detection and categorization very difficult. Effectively combating new malware variants requires the employment of novel techniques that are very different from conventional techniques. Deep learning (DL) methods, a type of traditional artificial intelligence (AI), are no longer sufficient to identify all new and sophisticated malware strains. A deep learning (DL) strategy, which differs significantly from conventional DL algorithms, may offer a potential answer to the challenge of identifying all malware types. This work suggests a brand-new deep-learning-based architecture that can categorise malware types using a hybrid approach. The study's key contribution is the suggestion of a novel hybrid design that optimally combines two diverse pre-trained network models. The design of the deep neural network architecture, the construction of the proposed deep neural network architecture, training of the proposed deep neural network architecture, and evaluation of the trained deep neural network comprise the four primary steps of this architecture. On the Malimg, Microsoft BIG 2015, and Malevis datasets, the suggested methodology was tested. The experimental results demonstrate that the proposed method outperforms the state-of-the-art methods described in the literature in its ability to

accurately and efficiently categorise malware. On the Malimg dataset, the suggested method demonstrated accuracy of 97.78%, outperforming the majority of DL-based malware detection techniques.

KEYWORDS: Artificial Intelligence, Neural network, Malicious software, CCNN.

INTRODUCTION

Malware is harmful software that pretends to be a legitimate program to infiltrate the computer. It is installed in different ways, but the most common are a phishing email, fake installer, infected attachment, and phishing links. Hackers make malware presentable to convince users into installing them. Often, the users are unaware that the program is malware because it looks legitimate. Basically, that's how malware gets installed on the computer. Once installed, malware hides in different folders in the computer. If it's an advanced type of malware, it can directly access the operating system. Then it starts to encrypt files and record personal information. To detect malware, the process malware detection is created. Malware detection is the process of scanning the computer and files to detect malware. It is effective at detecting malware because it involves multiple tools and approaches. It's not a one-way process, it's actually quite complex. The good thing is malware detection and removal take less than 50 seconds only. The number of reported security breaches due to virus, trojans, ransomwares, etc, has been growing considerably in recent years with reports of infections due to malware making the headlines, now more than ever. Almost every week one such security vulnerability is reported which may be seen as a failure by the security community on the control and detection of malicious content. Due to the significant number of malware attacks, and taking into consideration the increasing popularity and huge success of DL methodologies in classification in different domains, it is only natural to see these techniques applied to complement classical methodologies for malicious-content detection, in particular, supervised learning techniques.

MALWARE AND ITS TYPES

Malware Classification is the process of assigning a malware sample to a specific malware family. Malware within a family shares similar properties that can be used to create signatures for detection and classification. Signatures can be categorized as static or dynamic based on how they are extracted.

Distinguishing and classifying different types of malware from each other is important to better understanding how they can infect computers and devices, the threat level they pose and how to protect against them.

Types of malwares are:

Worms

Worms are spread via software vulnerabilities or phishing attacks. Once a worm has installed itself into your computer's memory, it starts to infect the whole machine and in some case of our whole network. Depending on the type of worm and your security measures, they can do serious damage. These parasitic nasties can

- Modify and delete files
- Inject malicious software onto computers
- Replicate themselves over and over to deplete system resources
- Steal your data
- Install a convenient backdoor for hackers

They can infect large numbers of computers fast, consuming bandwidth and overloading your web server as they go.

Viruses

Unlike worms, viruses need an already-infected active operating system or program to work. Viruses are typically attached to an executable file or a word document. Most people are probably aware that a .exe file extension could lead to issues if it's not from a trusted source. But there are hundreds of other file extensions that denote an executable file. Usually spread via infected websites, file sharing, or email attachment downloads, a virus will lie dormant until the infected host file or program is activated. Once that happens, the virus is able to replicate itself and spread through your systems. For computer viruses, your contact list is the equivalent of a packed train for the common cold. It hijacks your applications and uses your own apps to sneeze all over everyone sending out infected files to your colleagues, friends and clients. Because it looks like it's coming from a trustworthy source (you!), it has a much higher chance of spreading.

Bots & Botnets

A bot is a computer that's been infected with malware so it can be controlled remotely by a hacker. That bot (aka a zombie computer), can then be used to launch more attacks or to become part of a collection of bots (aka a botnet). Botnets are popular with hacker show-offs (the more bots you collect, the mightier a hacker you are) and cyber criminals spreading ransomware. Botnets can include millions of devices as they spread undetected. Botnets help hackers with all manner of malicious activities, including:

- DDoS attacks
- Keylogging, screenshots and webcam access
- Spreading other types of malware
- Sending spam and phishing messages

Trojan Horses

Just as it sounds, a Trojan Horse is a malicious program that disguises itself as a legitimate file. Because it looks trustworthy, users download it and hey presto, in storms the enemy. Trojans themselves are a doorway. Unlike a worm, they need a host to work. Once you've got the Trojan on your device, hackers can use it to

- Delete, modify and capture data
- Harvest your device as part of a botnet
- Spy on your device
- Gain access to your network

Ransomware

Ransomware denies or restricts access to your own files. Then it demands payment (usually with crypto-currencies) in return for letting you back in. In May 2017, a ransomware attack spread across 150 countries and compromised over 200k computers within just one day. Aptly named WannaCry, the attack caused damage estimated in the hundreds of millions to billions of dollars. WannaCry affected MS Operating systems that did not have the latest patch installed for a known vulnerability to reduce the risk of ransomware attacks.

- Always keep your Operating System up to date
- Keep your Anti-Virus software up to date
- Back-up your most important files
- Don't open attachments from unknown sources (WannaCry was spread via a .js attachment)

Adware & Scams

Adware is one of the better-known types of malware. It serves pop-ups and display ads that often have no relevance to you. Some users will put up with certain types of adware in return for free software (games for example). But not all adware is equal. At best, it's annoying and slows down your machine. At worst, the ads link to sites where malicious downloads await unsuspecting users. Adware can also deliver Spyware and is often easily hacked, making devices that have it installed a soft target for hackers, phishers and scammers.

Spyware

Spyware secretly records your online activity, harvesting your data and collecting personal information such as usernames, passwords and surfing habits. Spyware is a common threat, usually distributed as freeware or shareware that has an appealing function on the front end with a covert mission running in the background that you might never notice. It's often used to carry out identity theft and credit card fraud. Once on your computer, spyware relays your data to advertisers or cyber criminals. Some spyware installs additional malware that make changes to your settings.

Spam & Phishing

Phishing is a type of social engineering attack, rather than a type of malware. But is a common method of cyber attack. Phishing is successful since the emails sent, text messages and web links created look like they're from trusted sources. They're sent by criminals to fraudulently acquire personal and financial information. Some are highly sophisticated and can fool even your most savvy users. Especially in cases where a known contact's email account has been compromised and it appears you're getting an instruction from your boss or IT colleagues. Others are less sophisticated and simply spam as many emails as they can with a message about 'checking your bank account details'.

MALWARE CLASSIFICATION USING CUSTOMIZED NEURAL NETWORKS

Many anti-virus vendors and computer security researchers have worked hard against malicious hackers, still malware remains one of the most prominent digital cyberworld's threats. Motivated by the high return on investment ratio, the underground malware industry has gradually increased the sheer volume of threats on the internet each year. According to an AV-TEST report, by 2018, the total number of malware estimated that more than 800 million, which has risen 28 times over the last ten years. Even worse, viciously keen cybercriminals recently made considerable efforts to diversify their avenues of attacks . Inspired by the impressive increase in cryptocurrency values, 47 new families of crypto mining malware have appeared since early 2017 and have been linked to the 956% rise in the number of crypto mining attacks in the past year . Therefore, efficient, reliable, and scalable anti-malware solutions are essential to protect the trust worthiness of the digital cyber world. Several recent research efforts have shown that Deep learning(DL) offers a promising path for counteracting a large-scale malware attack. Such efforts were motivated by DL's great success in developing image classification accuracy, language translation, and a lot of other applications. However, there is a problem when analyze an appropriate classification scheme for malware based on the DL.

On the one hand, if we concentrate too much on identifying discriminative malware features to achieve high classification accuracy, the features are designed such that they lose their generality when they encounter a different operating environment. For example, when features are extracted from malware files' Program Executable (PE) headers, they are useful in classifying variants of PE malware belonging to

different families . A malware classification system equipped with these features cannot be used to detect signature less malware that only exists in the memory. On the other hand, if the features extracted from malware programs are too generic, such as the frequencies of n-gram byte sequences , they are hard to train a highly accurate malware classifier because of their lack of discriminative power . Convolutional Neural Networks (CNN) is the most efficiently functional, deep learning models for image classification in the present eras. The science of biology is an inspiration for CNN's multi-story architectures. High accuracy and rapid growth in CNN research have inspired us to use deep learning to classify malware images into different families. The proposed Customized Convolutional Neural Network (CCNN) modifies the basic structure of network architecture in such a way that CCNN recognizes and classifies malware images into distinct families and produces mucous membranes.

PROPOSED CUSTOMIZED CONVOLUTIONAL NEURAL NETWORK

The proposed Customized Convolutional Neural Network (CCNN) modifies the basic form of structurally-based design of the convolutional network so that CCNN acknowledges and classifies malware images into distinct families and produces much improved results than the earlier methodologies for malware classification. CCNN Model optimizes the number of layers instead of placing any layer restrictions. Also, a variety of filter sizes were used for intermediate convolutional CCNN layers. Leaky Rectified Linear Unit (Leaky ReLU) is used in the proposed architecture, which has no zero slopes in a negative direction. The schematic block diagram of the proposed malware classification scheme using CCNN is shown in Figure 3.1.

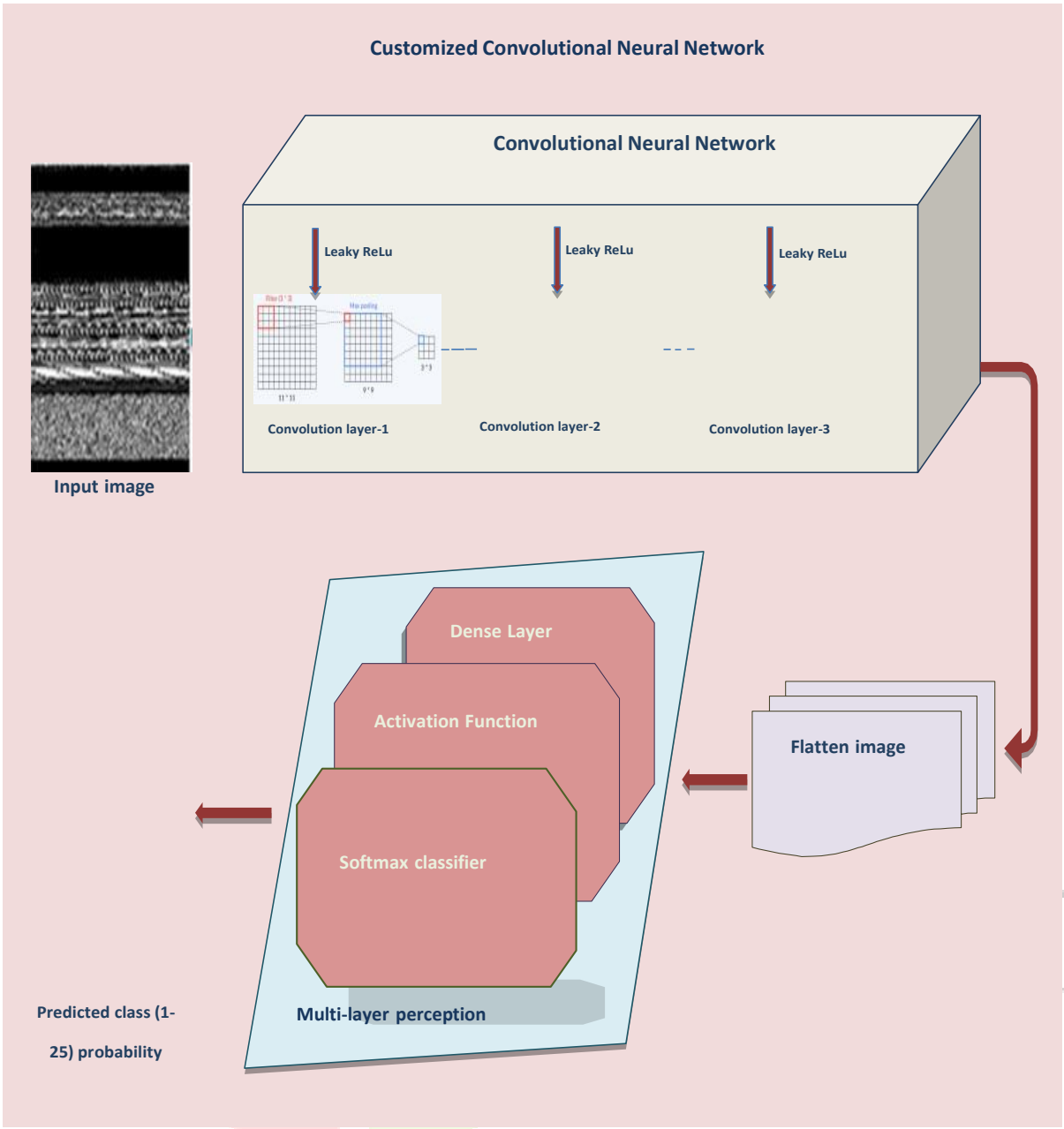

Figure 3.1: Schematic block diagram of CCNN

An input malware image is a grayscale image of 64x64x1, which is given as an input to CCNN. A Leaky ReLU activation function is used to situate non-linearity into theweights. A stride of two is made in max pooling to decrease the size of the output image produced from the convolution layers. In the second step, the image is flattened and transferred to the Multi-Layer Perceptron and a fully connected layer. The last stage is to use the Softmax classifier to return the image class probability for each class. A subset of neural systems is defined as the convents with mandatory models of accessibility among some layers. Typical neural networks tend to suffer from overfitting issues.

The customized convolutional neural network, on the other hand, exploits the localfeatures of the given malware images, such as treating pixels of malware images that are discerned nearby and inaccessible in an unexpected way. Therefore the suggested technique is used to distinguish the various malware images based on the picture's characteristics. The proposed CCNN architecture is shown in Figure 3.2, which has achieved enhanced results throughout the experimental assessment. Here the perceptive perception of the layer of CCNN is also displayed for the classification of malware images.

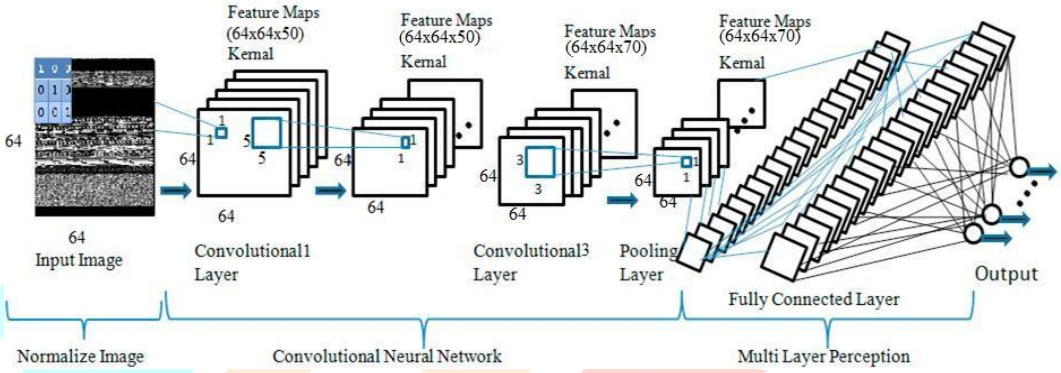

Figure 3.2: Overall architecture of CCNN

3.2.1 Convolution Layer

As shown in Figure 3.2, the CCNN input is a grayscale malware image $X^{w,h,d}$, in which the width is determined by w, depth is determined by d (d = 1), and the height is determined by h of the malware file. This layer is full for obtaining invariant malware image features, which are hierarchically and automatically cultured. Firstly, the convolution layer recognizes low-level features of an image by exhausting minor patches of the image to reserve spatial association between image pixels. It recognizes the characteristics of the local image from one layer and then maps them to the filters. For more details, a 2 Dimensional presentation of convolutional layers is discerned in Figure 3.3, in which the input malware image is of size 64 × 64 × 1. A (3 × 3 × 1) size filter is mapped to convert a given image to a feature map.

In Figure 3.2, 50 filters of size 5 × 5 × 1 are used in the first layer of convolution. The CCNN model takes a size 64 × 64 × 1 input malware image with p = 2, and zero padding. Filtering of the image is dot product of the filter and malware Image. This operation results in a volume of size 64 × 64 × 50 Where $w = h = 64 = {}^{(64-5)+2*2} + 11$ and d = 50 depth. The total number of neurons in this layer are 64 * 64 * 50 = 204800. In addition, a locally based region of the kernel is associated with every 204800 neuron. In the next two Convolutional Layers, 70 convolution filters of size 3 × 3 × 1 are placed and striding it among an input image with a strider 1. Initially, these filters are assigned with random values and then apply these filters on training data-set.

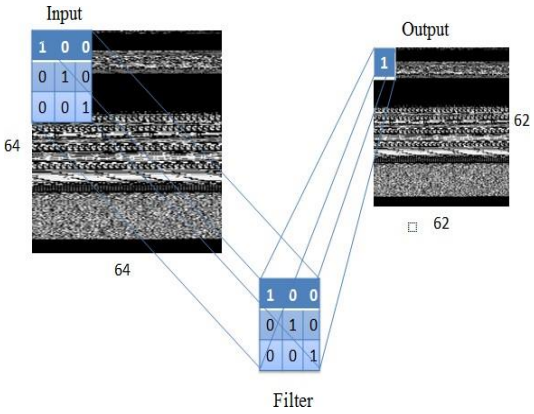

Figure 3.3: Feature map mirroring

3.2.2 Leaky ReLU Activation Function

Traditional CNN architecture uses Rectified Leaner Unit (ReLU) activation function to put non-linearity into the weights present in intermediate CNN strata. When there is a negative gradient involved, the ReLU activation function completely reduces it to zero, called as dying ReLU problem. In contrast, in the proposed architecture the Leaky ReLU activation function is used, which solves the dying ReLU problem, because there are no zero slopes present in a negative direction in Leaky ReLU. This function is used to mark out a distinct identification of likely characteristics. Thus, Leaky ReLU generates images of efficiency of$(64 \times 64 \times 50)$.

3.2.3 Pooling Layer

After the convolution layer, the input malware image is converted as an image stack. The next step is implementing the Pooling Layer; this is how we shrink the image stack, which is pretty straightforward. To reduce spatial size of the input image, pooling is used.Pooling controls an over-fitting problem by reducing image feature dimensions andcalculations. Pooling builds the network invariant to trivial biases and alterations. Importantly, it assures to acquire a scale-invariant illustration of the malware image. The different types of pooling include Max, Min, and Average Pooling. In a proposed CCNN model, the max pooling is used. We started with a window size of 2x2 pixels. Pick a stride of 2 and move a window in stride across a filtered image, which is the output of an earlier convolutional layer and for each window to take maximum value. Thus, the pooling layer generates an output image of size 32x32x50. A visualization of the pooling procedure is shown in Figure 3.4.

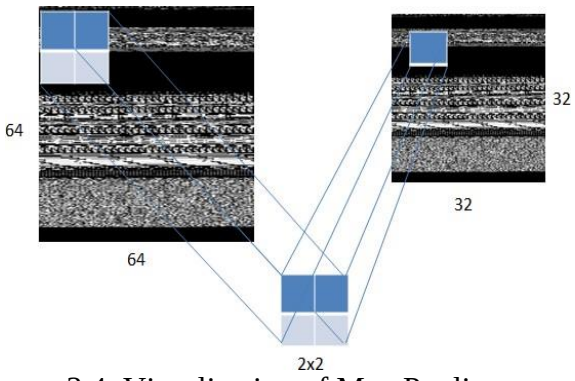

Figure 3.4: Visualization of Max Pooling

Output images of the Pooling layer are fed input to the second convolution layer. In this layer, seventy kernels of size (3 × 3 × 1) are used. All remaining parameters are kept same as the first convolutional layer. After applying the Leaky ReLU activation function, the second layer's output is (64 × 64 × 70). This is followed by max pooling operation for resampling the malware image. This procedure is repeated for the third convolutional layer with seventy kernels of size (3 × 3 × 1) and output volume of size (64 × 64 × 70) images are generated. Initially, all kernels are assigned a random value. In each epoch, kernel's value is changed for extracting the malware image features.

3.2.4 Fully Connected Layer

In this layer, neurons are associated with all activation of an earlier layer. Here, along with a bias offset, simple matrix multiplication is accomplished. Moreover, neurons are supposed when two or more related neurons discerned the same malware features frequently to ensure established co-dependency or co-adaption on all neurons, which introduces the problem of overfitting. In general, a dropout function is utilized in this kind of situation to eradicate such type of complexity by overlooking neurons random sets through the training phase. Dropout function shuts down nodes randomly so that learning or pattern based updates can be prevented. Subsequently, 256 dense layers are used in the proposed CCNN.

3.2.5 Softmax Classifier

Softmax classifiers provide the likelihood of increasing class numbers; thus, loss of the hinge gives the margin. As humans, it is far simpler for us to perceive probabilities than marginal scores. However, for datasets like Image Net, we also look at the rank-5 precision of Convolutional Neural Networks (where we search to see whether the ground-truth label for a given input image is in the top-5 predicted labels returned by a network).

In addition, due to the randomness in the primary weight, changes in weight should be done systematically, and subsequently the CCNN model is used. Note that every k output neuron communicating with all of the earlier layer's neurons indirectly relates to the family of malware. The

number of output neurons must be noted, and the number of malware families must be equal.

CONCLUSION

Despite extensive research on malware detection and categorization, detecting malware variants successfully remains a severe problem in the cyber security area. The identification of malware is difficult due to code obfuscation and packaging techniques. This paper offered a new deep learning architecture for detecting malware variants effectively. A hybrid architecture is proposed in the proposed architecture approach. This method relies on the transfer learning method and contains numerous exhaustively pre-trained networks. Initially, malware data was gathered by combining multiple large databases. The features are then retrieved with the help of pre-trained networks. Finally, a supervised learning method is used to execute the training phase of deep neural network design.