PSYCHOLOGY AND EDUCATION (2021) 58(5): 5036-5056
ISSN: 1553-6939

Optimized Convolutional Neural Networks based malware detection

Karrar Ahmed Kareem, Ethar Sabah Mohammad Ali

DhiQar Oil Company, Ministry of Oil, Iraq

Email: kalknany526@gmail.com

Abstract

Malware is a malicious code which is developed to harm a computer or network. The number of malwares is growing so fast and this amount of growth makes the computer security researchers invent new methods to protect computers and networks. It is a very serious problem and many efforts are devoted to malware detection in today's cybersecurity world. To detect malware, most antivirus scanners use a combination of signature matching and detection based on exploration. The obvious problem with this is that only known and lesser-known specimens can be identified based on the artifacts extracted from malware analysis. This is no longer the case due to the rapid growth of malware in recent years. In this thesis, we have proposed an optimized malware detection method for the Android platform using an optimized approach based on optimized neural networks, which are both a combination of permissions associated with risk and vulnerable API calls. And uses them as features of the CNN algorithm. The results show that the accuracy of the proposed method is 93.75%, which is 2.92% better than the basic article method.

Keywords: Malware Detection, Deep Learning, Machine Learning, Security, Classification, Convolutional Neural Networks, Particle Swarm Optimization (PSO).

1-1- introduction

Since the computers and internet are becoming increasingly universal, internet became an essential issue in daily life. ITU (international telecommunications union) reported that the number of internet users who always use the internet services such as electronic banking, electronic business, quick communication, training and entertainment in all over the world receives to 2.4 milliard until 2014. Exactly like the physical world, there are people with destructive intents (means cybercriminals) on the internet. They try to seize the benefits of legitimate users in their benefits. Malware (abbreviation of destructive malwares) is a general term which is widely used to express different kinds of unwanted software applications. These applications include virus, worm, Trojan, spy software, robots, Rotic, ransomware and the other cases [1]. Malwares are the most important threat for the internet and distributed systems. The battle between security analyst and malware researchers is a standing issue with the innovation growth simultaneously. Malware is an application which makes your framework suitable in order to do something

which needs an aggressive to do that. The simplest way to identify the malwares is to develop a consonant way to contrast with the suspicious codes recognition [2].

Numerous malware attacks cause serious security threats for the internet users. In order to protect the legitimate users from these threats, anti-malware software products are provided from different companies such as Comodo, Kaspersky, Kingsoft and Symantec. Commonly, signature-based method in these tools is widely used to identify the malwares in order to detect the various threats. Signature is a short sequence of bites which is often unique for each known malware and allows to be correctly identified new files with a small error [3]. However, due to the economic advantages, the malware writers quickly developed the development toolbox of automatic malware. These toolboxes use the techniques such as instruction virtualization, packing, polymorphism, imitation, metamorphosis for writing and changing the destructive codes. These toolboxes of creating malware greatly reduce the obstacles of rookie aggressive for entering the cybercrimes world (allow the inexperienced aggressive to write and customize their malware samples) and due to the simple availability cause the extensive reproduction of new malware samples. Therefore, malware samples will be quickly outbreak and quickly pollute the computers in all over the world. With the development of writing and creating destructive software techniques, the number of samples in the gray list is continuously increasing (for example, above 500000 new samples are registered each day by Kingsoft Cloud security center). So, smart methods which are gathered from new documents for automatic detection of malware applications on the cloud side are an urgent need. Therefore, many researches are reported about using data mining and machine learning techniques for developing the smart malware detection systems [4].

1-2- Problem statement

Malwares which are entered the victim's computer by hackers (aggressive) through security weaknesses of operating system or software program, are able to affect the normal performance, gather the sensitive information and steal the superior points of a user in order to do the destructive operations. Generally, main malwares include destructive subtitles, taking advantage of vulnerability, back doors, worms, Trojan, spywares, root kits and so on and also the combinations or the variety of the above types [5]. Yearly reports of the anti-virus companies show that thousands of new malwares are created every day. This new malware is becoming complicated that this is not possible to use traditional detection techniques such as signature-based detection searches to be able to detect the expectation of a special kind of malware for certain bites sequences in an object. Its drawback is that it cannot identify the zero-day or new malware because the destructive software signatures are not supposed to be in signature database [6].

In fact, malware recognition is a problem of software classification. The process of analysis and classification of software includes two main steps. First, different definitions of a program are made through

program analysis and some features are extracted from the program definitions. Then, it is possible to create classifiers to do the classification based on these features. Due to the number of programs which need to be processed, software automatic analysis and its classification is necessary. It is optimal that in each step, a little attempt or no attempt of human is needed [7]. In order to recognize the malware, most of the anti-virus scanners use the combination of signature adjustment and exploration-based recognition [8]. Clear problem of this issue is that only the known or less known samples based on the extracted artifacts are recognized by malwares analysis. Due to the fast growth of malware samples during the last years, this case is no longer answerable.

Deep learning is called because it is a kind of machine learning which imitates the human brain. Deep learning, which means teaching the way of thinking to the computer, is put aside from the research topics because it takes long time for data learning. However, because Alex net used CNN, a kind of deep learning, he won in an image recognition algorithm in 2012 with 16.4% error, deep learning is taken into consideration again [9]. Deep learning mechanisms facilitate feature extraction with extracting raw data as an input. They are generally classified into two kinds (1) convolution neural network (CNN) and (2) repeated neural network (RNN) [10].

In this thesis, we propose a malware recognition scheme for android platform using an optimized CNN-based approach which is a combination of risky permissions and vulnerable API calls and uses them as the CNN algorithm features.

1-3- challenges

- The most common method for malware recognition is the signature-based method. Signature is sensitive to the partial changes in destructive code. They need the previous knowledge about malware samples. Signature-based approaches cannot recognize the malwares which are modified or not seen before. Therefore, one of the main problems that anti-viruses are face is that how we can see the destructive code which is not seen before.
- In the field of malware recognition, gathering data samples is a relatively easy work. However, the determination of that whether an executive case is destructive takes a lot of work. So, obtaining the great amount of labeled data is difficult for us.
- The way of creating PSO-CNN-based malware recognition system
- Whether it is possible to use non-labeled data for malware recognition improvement
- Whether deep representation created by using PSO-CNN is useful for feature extraction and dimension reduction.

1-4- research goals

- The goal of using PSO-CNN as classification is to reduce feature vector dimensions and malware recognition.

1-5- thesis structure

The organization of thesis contents is that in section 2, first we describe the initial terms, in section 3 we overview the previous works in the field of malware recognition based on deep learning algorithms, in section 4 we describe the proposed solution, in section 5 we intend the proposed algorithm results and evaluation and at the end conclusion and suggestion is provided for the future works.

2-1- malware

One of the most important threats in the internet is the destructive software that often named as malware. Malware emerge at the wide range of changes and shapes such as viruses, worms, bot net, rot kit, Trojan horses and services of deny tools. For development, malwares take advantage of software vulnerability in browsers and operating systems or use the social-engineering techniques for tricking users in performing the destructive code. Usually, an anti-malware company receives thousands samples from new malwares every day. This sample is sent by the users who observe the suspicious codes on their systems or provided by the organizations which gather the malwares. The ability of classifying malware samples effectively and automatically named as behavioral clustering in the form of families with similar behavioral features [11].

2-2- malwares' classification

Based on different goals reproduction methods, malwares can be classified into different classes. This section is briefly provided about the common kinds of destructive software such as viruses, worms, Trojan, spy software, ransomware, robots ant rot kits.

Viruses: virus is a kind of code which can add itself to the other system programs and during execution the affected areas are contaminated. Viruses cannot be implemented separately because they should be activated by the "host" programs. Creeper virus which is written by Bob Tuma was a self-reproduction experimental program that explored in early 1970 for the first time.

Worms: against virus that needs to implement the "host" program in order to be activated, worm is a program which needs an independent implementation. Be aware that a worm can transfer a fully activated version of itself to the other devices. Morris worm is the first sample of this program which showed a manner like a worm. During the Morris appeal court, it is reported that based on the estimation, the cost of Morris worm elimination was about 100 million dollars. The other unpopular worms like Love Gate, CodeRed, SQL Slammer, MyDoom, StormWorm successfully attacked to 10 million windows computers and begot a lot of damages. For example, at the first day of publication, Code Red worms contaminated about 359.000 hosts on the internet, while MyDoom worms reduce 10% of the global access to the internet and cause to reduce the access of some websites about 50%.

Trojan: in comparison to worm which can transfer its fully activated version to the other machines, Trojan is software which pretends to be useful but performs its

destructive operations inwardly. One of the recent significant Trojans is Zeus (that is also named as Zbot) and is able to perform a lot of malicious and criminal acts. Zeus is often used to steal information on banking via entering the system. In June 2009, Prevxsecurity company discovered that above 74000 FTP accounts are endangered in many companies websites (such as ABC, Amazon, BusinessWeek, Cisco, NASA, Monster.com, Oracle, Play.com, America bank).

Spyware: spyware is a kind of destructive program which spies on users' activities without the user's notice and consent. Aggressive can use the spy software to observe the user activities, gathering Keystrokes and sensitive data removal (such as user logins, account information).

Ransomware: ransomware is one of the most popular destructive software in recent years which is installed hidden on the victim's computer and does an encryption attack which has negative effect on it. If the computer is contaminated to this malware, the victim is wanted to pay a ransom to aggressive for encryption.

Scareware: scareware is a new kind of destructive files which created to trick user in shopping and loading the unnecessary and dangerous software like fake anti-viruses that cause intensive economical threats and privacy violation for the victims.

Bots: bot is a destructive program that allows the owner of a bot to control the contaminated system remotely. Common methods of bot development are taking advantage of software vulnerabilities and using social-engineering techniques. When a system is contaminated, the owner of a bot can install worms, spywares and Trojans and converts the victim systems of person to a botnet. Botnet is widely used in running the distributed denial of service attacks (DDOS) and sending the fishing and spam emails. Agobot and SDbot are two of the most unpopular bots.

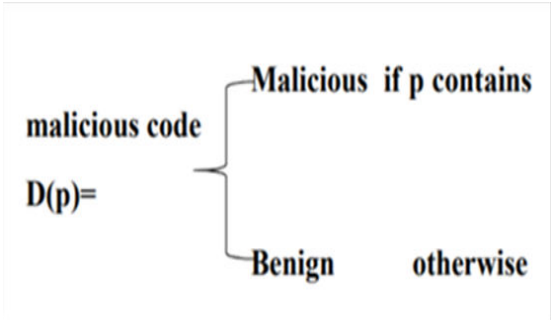
Rootkit: a rootkit is a kind of hidden software which is designed to hide some processes or programs and creating the possibility to have continuous access to the computers. Rootkit techniques are usable in different levels of a system; they can intervene in the calling of the applied programming interface (API) in the user mood or manipulate the operating system structures as a device driver or core module.

Combined malware: combined malwares combine two or more other shapes of destructive codes in a new kind to obtain the performances of the stronger attack and are able to be potential disturbances for the computer users such as spamware, Adware and similar cases. In fact, these destructive software are not usually unique. On the other hand, a kind of special malware may belong to some other malwares simultaneously.

2-3- Malware recognition

Malware recognition techniques: we can consider a malware recognition program "D" as a function which works in a domain contains an application set "P" and a set of malignant and benign programs. "D" analysis "P" programs which belong to the application set of "P" that whether this is a

malignant program (a normal program) or a malware (a destructive program) [13].



Recognition techniques: these are all techniques which are used in malware recognition. They can be classified into: signature-based recognition, exploratory-based and attribute-based method recognition.

2-3-1- signature-based recognition

Today, a lot of anti-virus software are using signature-based recognition technique. This method includes a kind of informational bank of signatures and traces the malwares with comparison of the existing pattern against database [14]. Signatures are usually a sequence of bites in software code to recognize the whether the scanned program is a malware or not [15]. Signature-based recognition is also known as a string scan which is the simplest form of a scan. Signature-based recognition can be dynamic, static or combinatory [17]. Dynamic signature-based recognition is specified using the information of program execution time for making decision about its destructiveness. Dynamic signature-based recognition follows the behavioral patterns that show the real destructive goal of a program. Static signature-based recognition examines the under-inspection program for the code sequences. The goal is accessing to a code which shows the program behavior in order to precisely determine the destructiveness of a program. Combinatory signature-based recognition is the combination of dynamic and static signature-based techniques [17]. Since this technique is based on the signatures informational bank, it is possible to precisely specify a lot of malwares which signatures are predetermined. This can be the most important advantage of signature-based recognition. Its implementation is also simple. Also, as it is mentioned in the introduction aggressive create new kinds of malwares every day that their signature may not be in the informational bank. Anti-malware motor may not be able to find the threat and this can begot a serious damage in the system. So, the authors in [15] briefed the below problems relevant to this method:

- Extraction and distribution of the work signature is complicated.
- Signature creation contains a manual intervention and needs the analysis of code.
- Signature can be easily compassed, especially when a new signature is created.
- The capacity size of the signature is growing at a worrying speed.

2-3-2- Recognition based on the behavior and exploratory method

On the contrary to the signature-based methods which search based on the known signature, exploration-based methods analysis the malware behavior. It means that, the abnormal and normal events are used to determine whether the performance

of a running process introduces it as a malware or not [16]. The authors in [18] introduce the below components for the behavior tracer:

- Data collection: this component collects the dynamic and static information.
- Interpretation: this component converts these raw information gathered by a module of data collection to the intermediate representations.
- Matching algorithm: this component is used to compare the presentation with the behavior signature. Figure 2-1 shows the behavioral revelation work.

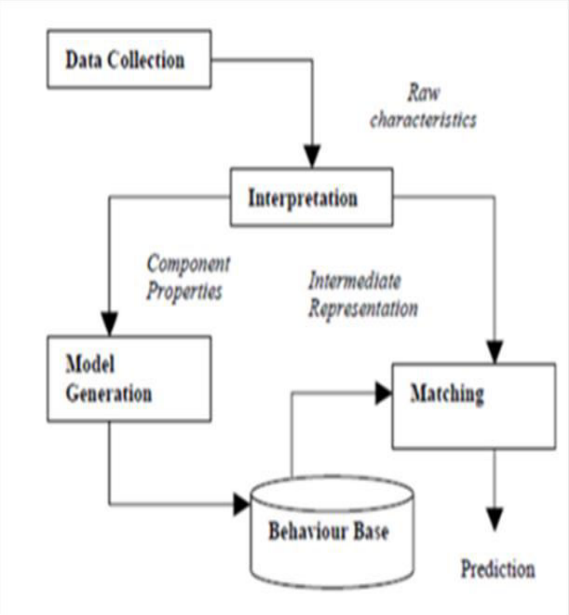


Figure 2- 1- Behavior detector [18]

Some of the computer security scientists like (14) say that exploration-based recognition is also known as behavior-based recognition. However some others like [16] believe that there is a little difference in their performance. Anti-malware based on the exploratory method examines the code itself and tries to match that with a known malware for exploring new kinds. Behavior-based method analysis the performances of a program then if the operation is done, it finds the malicious behaviors.

2-3-3- cloud-based malware recognition

To overcome the above challenges, most of the anti-malware sellers used the cloud-based (server) recognition. Cloud-based recognition workflow is provided in figure 5, this scheme can be defined below:

1. Users receive new files via internet through different channels on the side of customer.
2. Signature sets on the side of customers are first used by the anti-malware products in order to scan the new files. Files are labeled as "unknown", if they cannot be recognized by the existing signatures.
3. Unknown files information (such as files' credit, features of file or even files) are gathered and imparted to the cloud server.
4. In cloud server, classifier classified the samples of unknown files and exports the rules (benign or destructive).
5. Rule results are sent to the resources immediately.
6. Based on the results obtained from cloud server, it performs the scan process on the side of customer and then performs the recognition.
7. With a quick answer and feedback from the cloud server, costumer

users will have up-to-date security solutions.

Briefly, now the malware recognition is performed in the manner of server-costumer with the cloud-based architecture:

Blocking the invalid software programs from the black list and verifying the credit of valid software programs from white list on the side of customer (user) and predicting each unknown file (grey list) on the side of cloud (server) and quickly generating rule. Grey list includes unknown software files and these files can be malignant or benign. Traditionally, grey list is rejected or verified by the malware analysts. With the development of the writing techniques and creating malware, the number of file samples in the grey list is increasing consistently. For example, the grey list gathered by the security center Comodo Cloud or Kingsoft usually contains 500000 file samples in a day. Therefore, there is an essential need for developing smart techniques in order to support the malware recognition on the cloud side (server). In the recent years, business products like Kingsoft security products, Comodo anti-virus (AV), semantic anti-malware products (AM) and Microsoft explorer internet began to use data mining methods to perform destructive software recognition.

2-4- camouflage evaluation in malware

To understand and develop the malware recognition and analysis techniques, it is recommended to study malware camouflage. Malware camouflage is referred to hide the malwares in order to hide itself from the malware tracers as much as possible. There are a number of techniques which are used by the malware authors such as simple methods like encryption up to complicated and advanced methods like metamorphosis.

2-4-1- Encryption

Malware writers always want to write their plan in a way to not be recognized and taken into consideration by the malware detectors. Encryption is the simplest method for camouflage. This is the first technique for malware secrecy. This method includes encryption and decryption module. Each time the encryption is done with different keys while decryption is done with the same key. Since “unique” feature is not taken into consideration in decryption method, it is possible to be recognized. In year 1987, the first encrypted malware CASCADE is emerged. The structure of encrypted virus is shown in figure 2.2.

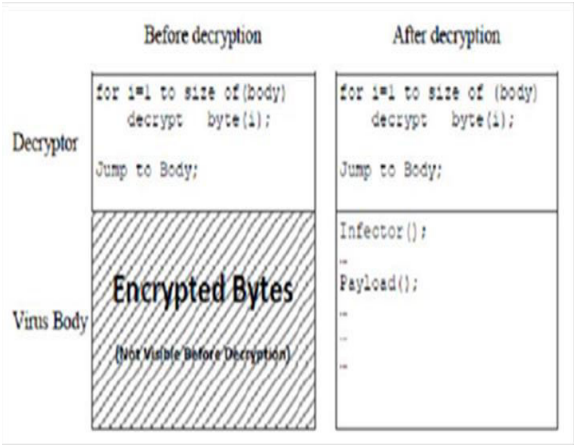


Figure 2-2- The structure of encrypted virus [19]

The main goal of this technique is preventing the recognition of anti-virus and static code analysis. This method also delays the researches’ process.

2-4-2- Oligomorphism

The first Oligomorphism virus was emerged in 1990 which is named as whale and was a virus of DOS kind. This virus is taken into consideration as a significant improvement in in malware camouflage. Also it is known as basic improvement in the encryption technique which is also known as the semi-polymorphism. Although Oligomorphism provides different encryption from an encryption list for each new attack, still there is chance to be recognized by an anti-virus with checking all decoders.

2-4-3- Polymorphism

In year 1990, the first polymorphism virus 1260 was created by Mark Washburn. Polymorphism virus is the combination of encryption and Oligomorphism but it is more complicated than other viruses. Therefore, its recognition by anti-viruses is too difficult because it changes its appearance by each version. The number of encryptions that can be created are not limited. These viruses use different blockage techniques to change its appearance. This change method is performed by a mutation motor.

2-4-4- Metamorphosis

Encryption is not a part of metamorphosis but in this generation the malware content will be changed. Its reason is the lack of need to a decoder. Also it implements a mutation motor like polymorphism but changes all its body not only its decoder. The basic idea is changing method in each new version while semantic is the same, it means that the appearance of virus will be changed but its meaning or performance will be fixed. The first metamorphism virus ACG was created for DOS in year 1998.

2-5- Convolutional neural network (CNN)

CNN is a kind of feed-forward in which connection pattern is inspired from visual cortex of animals among its neurons. CNN consists of three layers:

1. Fully connected
2. Convolution
3. Accumulator

All of the simple implementations of CNN can be introduced with below steps:

1. Convolution of several small filters on the input image
2. Sampling of this space with applying the activations of filter
3. Repeat steps 1 and 2 until you get to the up-level features
4. Apply standard feed-forward NN on the obtained features

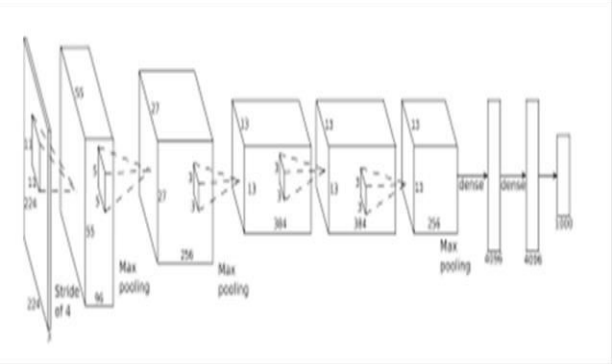


Figure 2- 3- Alexnet architecture [20]

Figure 2-3, convolutional architecture is used in Imagenet classification match. This architecture consists of 8 learnable layers, 5 convolutional layers and the others are fully connected layers.

1.Local connection

Connection of neurons to all neurons is impossible in the previous layer especially when colliding with the inputs with high dimensions like images because in these network architectures, the spatial structure of data is not taken into consideration. In CNN, each neuron is connected to a small area of input neurons (each neuron only connects to a small area of pixels in an input) and therefore CNNs are able to benefit from spatial local correlation with the implementation of a local connection pattern among the neurons of adjacent layers.

2. Convolutional layer

Convolutional layer includes a set of main cores of a CNN that is convolved in width and height of input features during passing forward and creates a 2D plan of a core. Briefly, a core consists of a layer of connection weights, input in the size of a small 2D patch and a single unit output.

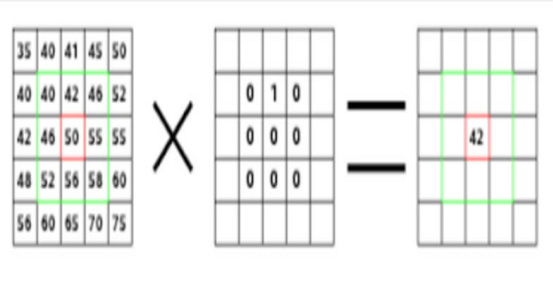


Figure 2- 4- Convolution [20]

Figure 2-4 shows that how a convolution works. Consider an image of pixels x55 like the figure in which 0 means the fully black values and 255 means the fully white values. At the center of the figure, a core of pixels x33 is defined. Each 8 cells are set on 0 except one which is equal to 1. Output is the result of core calculation in each possible situation in the image.

Step determines whether the convolution is applied on the core in all situations. For example, for step 1 it generates the typical convolution but for step 2 half of the convolutions are not considered because there should be 2 pixels distance among the centers. Output size after the core convolution with size Z on image N with step S is as below:

$$output = \frac{N - Z}{S} + 1$$

3.Integration layer

Integration is a kind of non-linear sampling. There are some non-linear functions for integration implementation such as minimum, maximum and average but the most common is maximum. Maximum integration works by dividing the image into set of rectangles without overlapping and generates the maximum amount for each sub-region.

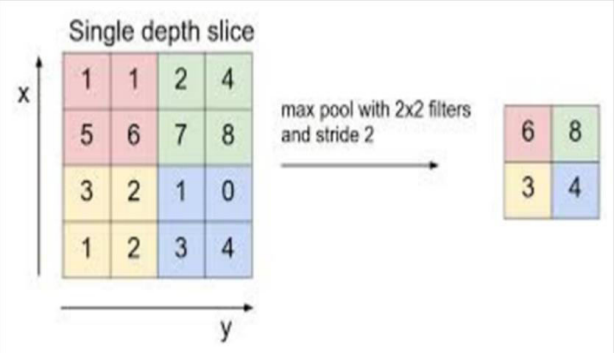


Figure 2-5- Max aggregator [20]

Key advantages of Max aggregator are:

1. Reducing the calculations for upper layers to eliminate non-max amounts
2. Provides a kind of unchanging conversion and more strength for the situation that presents a method to reduce middle representation dimensions.

RELATED WORKS

3-1- previous work

In this section, we study the researches done on malware recognition field based on deep learning algorithms.

In [21], a new model of 3-step combinatory android malware recognition is provided as SAMADroid. This model has three combinatory steps for analyzing and malware recognition. 1) static and dynamic analysis 2) distant and local host 3) machine learning intelligence. Two experiments are performed to determine the overhead performance arising from SAMADroid in real android device. In the first step, the performance of android device is observed through benchmark tools before and after the SAMADroid implementation. This experiment provides the overhead which is generated due to SAMADroid customer's program during the implementation of real android device. Second, it provides the SAMADroid overhead performance with MADAM. Results show that the overhead performance arising from MADAM is better than SAMADroid.

In [22], provide the machine learning method for android malware recognition which can automatically recognize the known and unknown kinds of destructive software if they belong to the kinds of analyzed malwares. In these methods, first they analyze the android program and create the control flow graph (CFG) from the resource code. Then, they extract the applied programming interface (API) inspired from CFG and create three different kinds of API datasets: Boolean datasets, Frequency datasets and Time datasets. Create three kinds of recognition: recognition model of using API, recognition model of API frequency, recognition model of API sequence based on datasets using machine learning methods. These experiments are performed on 10010 safe programs and 10683 destructive programs. Results show that recognition model obtained 98.98% recognition accuracy and has high integrity and stability. They analyze the correctness of each model using accuracy-metric standard classification criteria, calling and F score and compare the performance of these three models. By using the combinatory method, a collective model is created and obtained 98.98% recognition accuracy

In [23] a new method is provided which uses deep learning to improve the recognition of kinds of malware. Deep learning has shown a great performance in image recognition. To implement our proposed recognition method, they converted destructive code to black and white images. Then, images have been recognized and classified using a CNN which can automatically extract the image features of malware. Moreover, they have used bat algorithm to eliminate the lack of data balance among different malware families. In order to that the efficiency and effectiveness of the proposed method is verified, they performed some experiments:

1) to verify the efficiency in different methods of data equality 2) to examine the effect of different sizes of malware image and 3) to compare the proposed method results with the other results in destructive code recognition. Results show that this model has better accuracy and speed in comparison to the other malware recognition models.

In [24], introduced the principal license identification system (sigPID) and a system of malware recognition based on analysis of license use to overcome the rapidly increasing number of android malwares. They develop three levels of pruning with license data extraction instead of extraction and analysis of all android licenses to recognize the most important licenses which can be effective in discrimination among the safe and destructive programs. Results show that when a support vector machine is used as a classifier, it can obtain above 90% accuracy, calling, integrity and F score which is relatively similar to the cases generated by the main method while it does the analysis time 4-32 time lower than use cases of all licenses. In comparison to the other new approaches, SigPID is more effective with 93.62% recognition of the malware in dataset and 91.4% of new/unknown malware samples.

In [25], they proposed a deep learning-based method to recognize the malware internet of battlefield IOT (IOBT) through sequence of device operating code (opCode). In the first step, they transmit the opCodes to a vector space and use a special learning approach of a special space for the classification of safe and destructive programs. They also show the strength of our proposed method in the field of malware recognition and their stability against the attacks of useless code enforcement. A vector of selected features is created for each sample and the special space learning method is used for the malware classification. The evaluations on the paper show that our approach power in malware recognition is equal to integrity rate 98.37% and accuracy 98.59% and also has the ability to reduce the attacks of useless code interpolation.

In [26], a new framework is provided for android malware recognition. Framework uses different features to show the features of android applications in different aspects and the features are modified by using the extraction method based on existence or similarity for showing the effective features in malware recognition. A multiple deep learning method is proposed which is used as a malware recognition model. The author verifies the usefulness of this feature and their proposed feature vector generation method. Also the authors have proposed some experiments about classification use based on the unsupervised learning or ambiguous flexibility. Therefore, their framework was effective sufficiently to be used in android malware recognition.

PROPOSED METHOD

4-1- Introduction

In this section, we first introduce the overall structure of the proposed malware detection scheme and then we are going to describe each of the functions with details to show how this proposed scheme works for malware detection. Figure 4-1.shows the

general structure of the malware detection scheme.

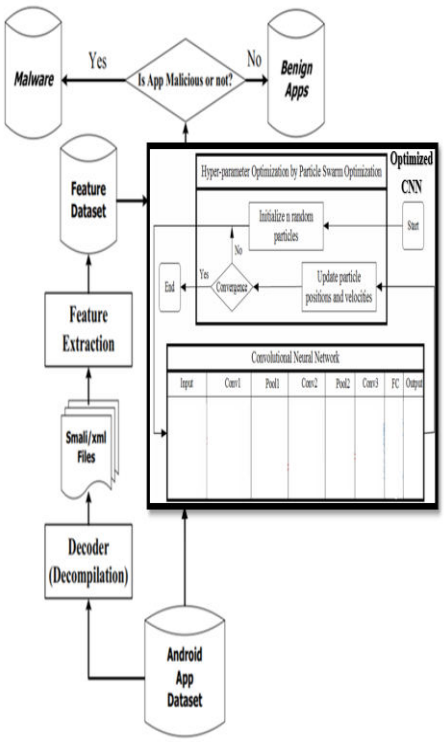


Figure 4-1- The diagram of general scheme of CNN-based malware detection

There are three main components in malware detection scheme, namely decryption (inverter translator), feature extraction, and classification. In the inverter translator component, each android application is not packaged and it is decoded in a small readable file. Some key features, such as risky permissions, suspicious API calls, and URLs are extracted in the extraction components due to several important and accepted measurements, such as TF-IDF and cosine similarities. Finally, we use a deep learning algorithm to build a classification model and evaluate them in the android application data set using their classification into malicious or secure programs.

4-2- Proposed method

In this thesis, we propose an automatic method of selecting a new hyper parameter to determine the optimal network configuration (hyper parameters) in deep neural networks using particle swarm optimization (PSO). The steps of the proposed method are shown in Figure 4-2.

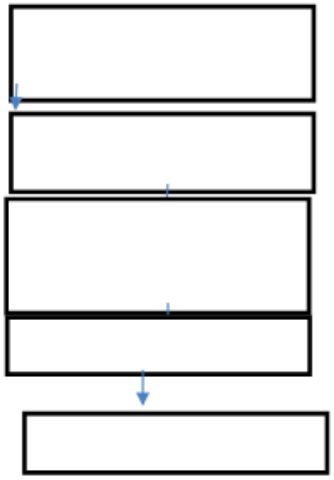


Figure 4- 2- Steps of the proposed method

4-3- Feature extraction by PCA

Principal component analysis (PCA) is a common feature extraction method in data science. Its purpose is to extract important information from the data set and to show these information as a set of new common variables called main components, and to represent the similarity pattern of observations and variables as points on maps [27]. Technically, PCA finds the special covariance matrix channel with the highest specific values and then it uses these to design data in a new subspace with equal or less dimensions. In practice, PCA converts a

matrix of n attributes into a new set of data (from hope) less than n attribute. It means that, it reduces the number of features by creating a new smaller number variable that records a significant part of the existing information in the main attributes.

4-4- CNN hyperparameters

The proposed method is an evolutionary algorithm for optimizing the parameters in CNN to classify malware. Deep networks have some parameters. One of those parameters is the size of the window. In convolutional networks, it works by inserting the image as the input. There are many filters, for example, let's imagine it is 5 * 5 or 2 * 2, where the filters move on the image to reach the end, and when it is placed on one part of the image, in that part of image, the inner product is done and this sliding window movement on image is called the convolution.

Now our goal is to have some of those windows that are equal to the number of filters. Second, we want to find out what should be their sizes in each layer? Third, there is another layer called pooling, because in the step of windows sizes, which are sometimes multiplied in the image and placed as the output of something, they should be converted into a smaller value, so that the number of parameters do not increase. This process is done by pooling. Pooling is such that the 2x2 window is placed on each part of that image and selects the largest number (maxpooling). Another feature of pooling is that it makes the network resistant to small changes. Therefore, pooling is important to us. So the number of parameters we have considered for optimization is 11, which are as follows:

1. For the convolution layer, we have three parameters: length (L), width (W) and number of filters (N).

2- In maxpooling, we have one parameter, the size of the window (S)

If we have a network that has three layers of convolution and two layers of maxpooling, with these parameters, the number of parameters becomes 11 (Figure 4-2). And it is good to get the optimal value of these parameters that we use the PSO method for optimization. CNN network architecture is in the Figure 4-3.

L1	W1	N1	S2	L3	W3	N3	S4	L5	W5	N5

Figure 4- 3- e Vector of a particle in evolutionary optimization.

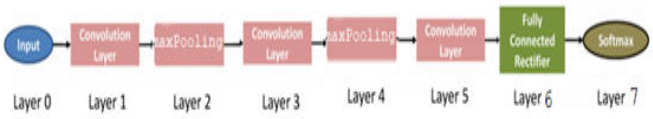


Figure 4- 4- High-level design of optimized CNN model

4-5- Particle swarm optimization (PSO)

PSO uses a number of particles that form a group movement in the search space to look for the best solution. Each particle is considered as a point in the next D-dimensional space that sets up its flight according to the experience of its flight as well as the flight experience of other particles. Particles fly at a certain speed in the next D space to find the optimal solution. The speed of the particle i is expressed as $V_i = (v_{i1}, v_{i2},...,v_{iD})$, the

location of the particle i is expressed as (x_{i1}, x_{i2},..., x_{iD}), the optimal location of the particle i is expressed as $P_i = (p_{i1}, p_{i2},..., p_{iD})$, which is called pbest. The optimal global position of all particles is expressed as Pg = (p_{g1}, p_{g2},...,p_{gD}), which is also called gbest. Each particle in the group has a function to calculate the fitness value. The formula for updating speed, from d-dimension in PSO standard, is shown in formulas (1.4) and (2.4) [28]:

$$v_{id} = w \times v_{id} + c_1 \times rand( ) \times (p_{id} - x_{id}) + c_2 \times Rand( ) \times (p_{gd} - x_{id}) \quad (1\text{-}4)$$

$$x_{id} = x_{id} + v_{id} \quad (2\text{-}4)$$

PSO parameters include: Q (population number), w (inertial weight), C1 and C2 (acceleration constant), v_{max} (maximum speed), G_{max} (maximum number of repetitions), rand () and random functions are with values in the range [0 , 1]. C_1 and C_2 values usually take a constant value of 3.

4-6- Optimized Convolutional Neural Networks (CNN)

Among several different DLs, Convolutional Neural Networks (CNNs) are currently the latest solution for complex classification problems, especially about image analysis. For the existing complexity, we propose an optimized CNN PSO classifier. Traditional CNN has 3 layers that have results and maximum integration. In special cases, the input tensor is a single vector with the selected data set features. Each i convolution layer can be described by 3 hyperparameters, respectively: Output channels, convolution window, and maximum integration window. In addition to the convolution layers, there is a fully connected rectifier layer for the output channels, which is a final software layer with random deletion. According to the meta-parameters, our goal is to find the optimal (or almost optimal) values for the target classification problems. To achieve this goal, we applied PSO-based optimization, and ultimately, after we trained CNN, we identified malware and safe samples with the best proposed PSO vector.

The neural network that we used in this study, has three layers of convolution, because the layer of convolution is two-dimensional. The goal is to convert this input image into a picture, and the number of dimensions is 330 and it is converted to 2×2. Each image is a data. In Figure 4-5, as you can see, 9 images are drawn. Because most of our images are zero and one, the tag on the title of each image is shown and most of them actually is zero and one. The on points are 1 and the off points are 0. In network training, we train the first layers of convolution based on PSO parameters. The activation function is relu.

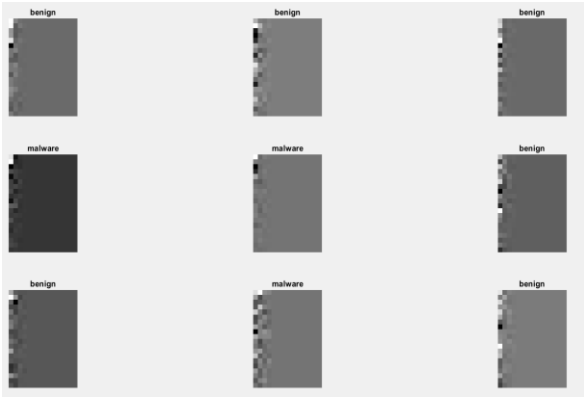


Figure 4- 5- Malicious and safe images in the data set

5-1- Dataset

In this thesis, both malware and safe samples were downloaded from stores such as android malware data collection, Kaggle. Data details are given in Tables 5-1. This data set is the result of research production in the field of machine learning and androic security. Data were obtained by a process involving the creation of a binary vector of permissions used, which is analyzed for each practical application (ie, 1 = used and 0 = unused). In addition, malware / safe samples are divided by "Type". 1 for malware and 0 for non-malware.

Table 5- 1- Description of data set

characteristic	Value
The number of features	300
The number of samples taken	398
number of malware samples in the data set	199
number of good samples in the data set	199

2.5. The initialization of parameters

The implementation process was carried out by the 2018b MATLAB in a 2.1 GHz Pentium seven-core with 12 GB of RAM. In order to check the quality of the proposed algorithm, we set the parameters according to the base paper parameters [25]. In this way that we considered a value of 3 for the number of convolutional layers in CNN, and the image dimensions were considered 20 ×15. Table 5-2 shows the settings for CNN parameters. Table 5-3 shows the settings for PSO parameters. Also, the percentage of data for network training is 80% and for network testing is 20%.

Table 5- 2- The initial values for parameters in CNN

parameters	Primary values
The number of convolutional layers	3
The image dimensions	[20,15,1]
numepochs	1000
Activation function	relu
Learning rate	0.01
BatchSize	50

Table 5-3- The initial values for parameters in PSO

Parameters	Primary values
The number of population	100
The number of repetition	20
SelfAdjustmentWeight	3
SocialAdjustmentWeight	3
The number of parameters	11
Upper bound	15
Lower bound	1

5-3- Evaluation metrics

To compare the proposed solution with algorithms such as neural network, we will use more accurate measurement parameters. One of the criteria used to show the precision of data classification in the CNN

algorithm is the method of finding the accuracy value and error rate and the F1 criterion.

The choice of a criterion for evaluating the efficiency of a method depends on the problem we are trying to solve. Suppose that a number of data samples are available. These data are given to the model individually and for each, a class is received as output. The class predicted by the model and the actual data class can be displayed in a table. This table is called the confusion matrix.

Table 5- 4- confusion table

		The label of predicted class	
	predicted real	Safe	malware
The label of real class	safe	*True negative (TN)*	*False positive (FP)*
	Malware	*False negative (FN)*	*True positive (TP)*

True Positive: the samples that have been correctly identified as malware by test.

False positive: the samples that have been identified wrongly as malware by test.

True Negative: the samples that have been correctly identified as safe by the test.

False negatives: the samples that have identified wrongly by the test.

5-3-1- Accuracy metrics

The ability of a test to correctly differentiate between healthy and unhealthy is called accuracy. To calculate the accuracy of a test, we should obtain the ratio of the total true positive and true negative samples to all of the tested items. Mathematically, this ratio can be proposed as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

(1-5)

5-3-2- Precision metrics

This criterion means the ratio of samples that are positively labeled and their class is really positive (Equation (2-5)).

$$Precision = \frac{TP}{TP + FP}$$

(2-5)

5-3-3- Recall metrics

It shows the efficiency of classifier according to the number of events in the class. In fact, the probable prediction is the nonexistence of the desired state by algorithm according to the Equation (3-5).

$$Recall = \frac{TP}{TP + TN}$$

(3-5)

5-3-4- F1 score metrics

The F1 criterion is an appropriate criterion for evaluating the accuracy of an experiment. This criterion considers precision and recall together (Equation (4-5)). The F1 criterion is one at best and zero at worst.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

(4-5)

5-4- The results evaluation

In this section, we evaluate the efficiency of classification and the generalization of the proposed approach in solving malware detection. In this experiment, two different methods were used to build the final classification models. The population size has set to 100 and the maximum number of PSO repetitions equals to 20. The proposed method is compared with 3d CNN [29] and 1-D CNN [30] methods and the evaluation results are shown in Tables 5-5.

Table 5- 5- Results of comparing the accuracy of the proposed method with the base paper

	Accuracy	Precision	Recall	F1 score
proposed method	93.75	90.95	93.78	93.58
base paper	90.83	90.77	90.85	90.80
3d CNN [29]	89	89	85	91
1-D CNN [30]	93.17	-	-	-

Accuracy is a general criterion for evaluating the efficiency of an algorithm for identifying both malware class and safe class. The proposed approach achieves high accuracy equals to 93.75%, while the base paper approach has 90.83% accuracy. Reminder or diagnostic rate is an important criterion and the proposed approach is obtained with 93.78% accuracy compared to the base paper method which is 90.85%.

Our proposed approach is also better than the base paper approach in terms of precision and F1 measurement. It seems that by choosing the right features, the useful features of the partial class are effective in the classification stage.

In addition, it also seems that the use of deep neural networks for classification leads to better classification. As shown in the results obtained in first row of the Table 1-5, it can be seen that the proposed approach can find a more suitable network structure with better hyperparameter configuration, which is then used to start and train the CNN classifiers and they achieve great performance in both the training phase and the final models (after training). Table 5-5.shows the superiority of the solutions created by the proposed approach. This indicates that the PSO algorithm, combined with deep learning architectures, can usually determine a more appropriate network.

6-1- Conclusion

The IoT, especially the IoBT, will become more important in the predictable future. There will be no perfect solution for malware detection, but we can be sure of a constant battle between cyber attackers and cyber defenders. Therefore, it is important that we keep the constant pressure on the threat actors.

The most common way for malware detection is through signature-based method. Signature is sensitive to minor changes in malicious code. They need prior knowledge of malware samples. Signature-based approaches cannot identify modified or previously unseen malwares. Therefore, one of the main problems that antiviruses face is how to see malicious code that has not been seen before. In the field of malware detection, data collection is relatively an easy task. However, for determining if an executive case is destructive or not, takes a lot of time. Therefore, it is difficult for us to obtain large amounts of tagged data.

Some key features, such as risky permissions, suspicious API calls, and URLs, are extracted in the PCA method due to several important and accepted measurements, such as TF-IDF and cosine similarities. At the end, we use a deep learning algorithm to build a classification model and evaluate them in the android application data set by classifying them into malicious or safe programs. Among several different DLs, Convolutional Neural Networks (CNNs) are currently the latest solution for complex classification problems, especially about image analysis. For the existing complexity, we propose an optimized CNN PSO classifier. The results show that the accuracy of the proposed method is 93.75%, which is improved 2.92% in comparison to the base paper method.

6-2- Future works

In the future, our goal is to examine our approach in the field of a larger and more extended set of data and also, we want to implement prototypes of the proposed method in a real-world IoT and IoBT system for evaluation and refinement. In addition, in order to take advantage of distributed computing benefits, the proposed method is going to have a kind of redesign of performance in a network of IoT nodes. We also intend to use metaheuristic algorithms for optimizing hyper parameters of CNN.

References

[1]. Ye, Yanfang, Tao Li, Donald Adjeroh, and S. SitharamaIyengar. "A survey on malware detection using data mining techniques." ACM Computing Surveys (CSUR) 50, no. 3 (2017): 1-40.

[2]. Souri, Alireza, and Rahil Hosseini. "A state-of-the-art survey of malware detection approaches using data mining techniques." Human-centric Computing and Information Sciences 8, no. 1 (2018): 3.

[3]. Ye, Yanfang, Tao Li, Shenghuo Zhu, WeiweiZhuang, EgemenTas, Umesh Gupta, and MelihAbdulhayoglu. "Combining file content and file relations for cloud based malware detection." In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 222-230. ACM, 2011.

[4]. Ni, Ming, Tao Li, Qianmu Li, Hong Zhang, and Yanfang Ye. "FindMal: A file-to-file social network based malware detection framework." Knowledge-Based Systems 112 (2016): 142-151.

[5]. Ni, Sang, Quan Qian, and Rui Zhang. "Malware identification using visualization images and deep learning." Computers & Security 77 (2018): 871-885.

[6]. Kumar, Rajesh, Zhang Xiaosong, RiazUllah Khan, IjazAhad, and Jay Kumar. "Malicious code detection based on image processing using deep learning." In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, pp. 81-85. ACM, 2018.

[7]. Nix, Robin, and Jian Zhang. "Classification of Android apps and malware using deep neural

networks." In 2017 International joint conference on neural networks (IJCNN), pp. 1871-1878. IEEE, 2017.

[8]. Bragen, Simen Rune. "Malware detection through opcode sequence analysis using machine learning." Master's thesis, 2015.

[9]. Lee, Yoon-seon, Jae-ung Lee, and Woo-young Soh. "Trend of Malware Detection Using Deep Learning." In Proceedings of the 2nd International Conference on Education and Multimedia Technology, pp. 102-106. ACM, 2018.

[10]. Vinayakumar, R., K. P. Soman, and PrabaharanPoornachandran. "Detecting malicious domain names using deep learning approaches at scale." Journal of Intelligent & Fuzzy Systems 34, no. 3 (2018): 1355-1367.

[11]. Kirda, Engin, H. Van Tilborg, and S. Jajodia. "Malware Behavior Clustering." (2011): 751-752.

[12]. Symantec. 2016. Internet Security Threat Report. Retrieved from https://www.symantec.com/content/dam/ symantec/docs/reports/istr-21-2016-en.pdf.

[13]. Imtithal A Saeed, Ali Selamat, Ali M A Abuagoub, "A Survey on Malware and Malware Detection Systems", International Journal of Computer Applications (0975-8887), Volume 67 – No.16, April 2013

[14]. JyotiLandage, M.P. Wankhade, "Malware and Malware Detection Techniques: A Survey", International Journal of Engineering Research and Technology, ISSN: 2278-0181, Vol. 2, 2013

[15]. Vinod P., V. Laxmi, M.S. Gaur, "Survey on Malware Detection Methods" Workshop on Computer and Internet Security, Department of Computer and Engineering, PrabhuGoel Research Centre for Computer and Internet Security, IIT, Kanpur, pp-74-7, March 2009

[16]. Sulaiman Al Amro, Ali Alkhalifah, "A Comparative Study of Virus Detection Techniques", International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 9, No.6, 2015

[17]. NwokediIdika, Aditya P. Mathur, "A survey on Malware Detection Techniques", February 2007

[18]. Kim, Chang Hoon, Kabanga E. Kamundala, and Sinjae Kang. "Efficiency-based comparison on malware detection techniques." In 2018 International Conference on Platform Technology and Service (PlatCon), pp. 1-6. IEEE, 2018.

[19]. Tahir, Rabia. "A study on malware and malware detection techniques." International Journal of Education and Management Engineering 8, no. 2 (2018): 20.

[20]. Gibert, Daniel. "Convolutional neural networks for malware classification." University Rovira i Virgili, Tarragona, Spain (2016).

[21]. Arshad, Saba, Munam A. Shah, Abdul Wahid, AmjadMehmood, Houbing Song, and Hongnian Yu. "Samadroid: a novel 3-level hybrid malware detection model for android

operating system." IEEE Access 6 (2018): 4321-4339.

[22]. Ma, Zhuo, HaoranGe, Yang Liu, Meng Zhao, and Jianfeng Ma. "A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms." IEEE Access 7 (2019): 21235-21245.

[23]. Cui, Zhihua, FeiXue, XingjuanCai, Yang Cao, Gai-ge Wang, and Jinjun Chen. "Detection of malicious code variants based on deep learning." IEEE Transactions on Industrial Informatics 14, no. 7 (2018): 3187-3196.

[24]. Li, Jin, Lichao Sun, Qiben Yan, Zhiqiang Li, WitawasSrisa-an, and Heng Ye. "Significant permission identification for machine-learning-based android malware detection." IEEE Transactions on Industrial Informatics 14, no. 7 (2018): 3216-3225.

[25]. Azmoodeh, Amin, Ali Dehghantanha, and Kim-Kwang Raymond Choo. "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning." IEEE Transactions on Sustainable Computing 4, no. 1 (2018): 88-95.

[26]. Kim, TaeGuen, BooJoong Kang, Mina Rho, SakirSezer, and EulGyuIm. "A multimodal deep learning method for Android malware detection using various features." IEEE Transactions on Information Forensics and Security 14, no. 3 (2018): 773-788.