International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Malware Classification Using Deep Learning

K V Sreelakshmi[1], Dileesh E D[2]

M Tech Student, Department of Computer Science Engineering, Govt. Engineering College, Idukki, Kerala, India[1]

Assistant Professor, Department of Computer Science Engineering, Govt. Engineering College, Idukki, Kerala, India[2]

ABSTRACT: Malware code attacks have rapidly increased with the growth of the Internet and have become a major threat to Internet security. Detecting malware is necessary for the protection against data theft, security breaches, and other dangers. The proposed method uses deep learning for classifying malware into their respective classes. The malware binary is converted into RGB images. Then the images are identified and classified using Convolution Neural Network (CNN), which extracts the features of malware images automatically. CNN requires fixed-size input images for training and testing. The malware images come in variable size, thus cropping and warping of the image to make it into fixed-size may lead to information loss thus resulting in low recognition accuracy. Hence an additional pooling layer called Spatial Pooling Layer (SPP) is used in CNN to eliminate this constraint of fixed size input image resulting in improvement in the accuracy of the model.

KEYWORDS: Malware classification, Convolutional Neural Network, SPP-net, Deep Learning, Visualization Approach.

I. INTRODUCTION

Internet usage has become an integral part of our daily lives. The browsers download different kinds of software from different websites. One of the main drawbacks of widespread usage of the Internet is that many computers are prone to security attacks and can get infected with malware. Malware is a software or program that has a malicious objective. Malware tries to gain access to a computer system or network to steal private data in an unauthorized manner. These are usually classified according to their propagation method and their actions that are performed on the infected machine using the designed malicious program. Some of these malware are Adware, Botnet, Ransomware, Rootkit, Spyware, Trojan, Virus, Worm, etc. If an Internet user surfs the World Wide Web without using any anti-virus or firewall protection, then it would be enough for him to understand the risk of malware attacks and its consequences. San Diego Supercomputer Centre (SDC) experimented to determine the probability of being attacked by malware while surfing. For this experiment, they installed a Red Hat Linux 5.2 without any security set-ups and then connected the computer to the Internet. Within 8 hours after the installation, the computer was hacked. In 21 days, the computer had undergone 20 attacks and after 40 days of installation the computer was regarded as "compromised". A report from Symantec showed that in 2016 over 401 million malicious codes were found including 357 million new malicious code variants. The appearance of malware in mobile and IoT devices have grown rapidly. To date, 68 new malware families and more than 10000 malware have been reported. With sufficient information about potential malware, malware detection is the key area of concern for researchers and the general population. With the ever-increasing risk of attacks, malware detection and classification has become a crucial problem in the field of cyber security. The onus lies on the security researchers to determine new mechanisms against malware.

To devise new defense for protecting the vulnerable systems from malware attacks, the malware analysis is done to understand the mode of operation and impact on a system of a given piece of malware code. Two types of analysis techniques are Static analysis and Dynamic analysis. Static analysis techniques involve tools to parse the executable and extract features for analysis. Whereas dynamic analysis techniques require the malware to be run in a virtual environment to observe its behavior. These analysis techniques can be easily compromised by hiding malware codes that have to be run in a virtual environment. So, there has been a need to explore new approaches to analyze the malware. Traditional malware identification tools used signature-based matching for identifying malware. The security companies analyze samples and generate a unique signature for those samples. These signatures were then updated in a database and using this database the signature of every file in the machine was compared. If the signature of a file matched the signature in the database then the file was labeled as malware. As the number of files to be analyzed by the security companies increased, so doing the analysis manually is not possible. So the process of analysis needs to be automated to reduce the number of files that require manual analysis.

This paper proposes a malware classification model using a visualization approach for representing malware binaries and to determine whether there is any pattern visible in that visual representation. Then using this representation, the

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

malware samples are classified into their families. Malware classification is very essential as it helps an analyst to understand the malware behavior. This helps the analyst to determine new detection and sanitization techniques for malware by getting information about the malware family or class. Here the malware samples are represented as RGB images and are classified using Convolutional Neural Network (CNN). By converting the malware into images, we have converted the analyses process free from the problems faced by static and dynamic analysis. All the CNN models have a major drawback that it accepts fixed-size images for training and testing. As malware files come in variable sizes, thus modifying an input image to fixed size is not ideal in case of malware classification as it is susceptible to content loss before the classification stage. To overcome the constraint of fixed size image, an additional pooling layer called Spatial Pyramid Pooling (SPP) which eliminates these fixed-size constraints. Thus the new network can generate fixed-length output for variable size images.

The remainder of this paper is structured as follows: Section II gives the details about the related researches, Section III details the proposed system, Section IV gives the details of the experimental setup, Section V shows the result analysis and Section VI concludes the paper.

II. RELATED WORKS

This section presents previous researches based on static analysis, dynamic analysis, hybrid analysis, and visualization approaches for malware classification and detection.

A. STATIC ANALYSIS APPROACHES

Detection of malware using static analysis requires the malware binary to be disassembled and analyses its execution logic. This type of detection method cannot detect new malware. Malware detection based on static analysis is faster and has higher accuracy than dynamic analysis. The drawback is that it is very time consuming and requires an expert to perform analysis.

Zhao et al. [1] proposed a virus detection system where the malware executable is converted into assembly language using a disassembler. Then this assembly language is converted into a control flow graph. From the control flow graph, the features were extracted using Rapid Arithmetic Method, and these features are trained and classified using Decision tree, Bagging, and Random Forest algorithms.

Ye et al. [2] proposed a system for malware detection using Object Oriented Association (OOA) mining-based classification. The program executable was converted into API execution sequences using a PE Parser. The API sequences were then grouped into a 32-bit ID which represented the corresponding API functions. The API calls were used as the signature for the PE files. Using the OOA mining algorithm class association rules were generated which was then used for classification. They demonstrated using experimental analysis that the IMDS model had higher accuracy and efficiency than the popular anti-virus software like Norton antivirus, McAfee Virus Scan, as well as previous data mining based detection systems which employed Naive Bayes, Support Vector Machine (SVM) and Decision Tree techniques.

B. DYNAMIC ANALYSIS APPROACHES

In the dynamic analysis, the malware codes are run in a virtually controlled environment and the behavior of the malware is analyzed during its execution. During the execution, its system interaction effects on the machine are observed. It is time-consuming and some malware may not be able to run in such a virtual environment. It can detect known and unknown malware but has a high false-positive rate.

Zolkipli et al. [3] proposed an approach for behavior-based analysis of malware and to classify malware into new classes using artificial intelligence techniques. They used Honeypot, HoneyClient, and Amun to collect malware samples for analysis. Behavior reports for each sample were generated using CWSandbox and Anubis virtual machines and each report was analyzed manually. Then using the Artificial Intelligence technique the malware samples were grouped as Worms and Trojans. The only drawback of this approach was that they did not automate the report analysis, so considering the huge volume of malware samples generated in present times, a human analysis will not be feasible.

Chun et al. [4] proposed a malware detection system in which the API logs are collected by running the malware samples in a virtual environment. From the API Log collected the features for classification are extracted. The classifier used is Naive Bayesian, J48 Decision Tree, and Support Vector Machine to classify whether the given sample is a malware or benign. The drawback of this system is the monitoring overhead and tracing of API is less accurate.

Tobiyama et al. [5] proposed a malware detection system based on traffic analysis. To generate the dataset the malware files were run in a controlled virtual environment such as Cuckoo Sandbox. The malware behavior was traced to find generated and injected processes. Log files are created about the behavior of the malware

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

analyzedduring runtime. Features are extracted and converted into images and then given to CNN for classification. The drawback is that it used a small dataset and the malware can detect the presence of the virtual environment, so it may behave differently resulting in a high false-positive rate.

Biley et al. [6] developed a classification model that defines malware behavior as a system state change. The malware binaries were executed in the virtual environment with Windows XP installed. The virtual machine is partially secured using a firewall to prevent any immediate malware attack behavior during the execution period. Then a behavioral fingerprint of malware's activity is created which includes files written, processes created and network connection, etc. To cluster the malware a pairwise single-linkage hierarchical clustering of the fingerprints using Normalized Compression Distance (NCD) as a distance metric was used. This technique was applied to automated classification and analysis of 3700 malware samples collected over a period of six months. They also measure and compare the consistency, completeness, and conciseness of the clusters with that of AV products. The drawback of this work is that the capability and environment of the virtualized system are static throughout the experiments for consistency.

C. HYBRID ANALYSIS APPROACHES

Anderson et al. [7] proposed a method, in which multiple data sources (the static binary, the disassembled binary file, its control flow graph, a dynamic instruction trace, and system call trace, and a file information feature vector) are used. For the binary file, disassembled file, and two dynamic traces, kernels based on the Markov chain graphs are used. The Graphlet kernel was used for the control flow graph and the standard Gaussian kernel was used for the file information feature vector. To find the weighted combination of the data sources multiple kernel learning was employed and Support Vector Machine was used to classify the dataset into malware or benign. The system tested 780 malware and 776 benign instances.

Islam et al. [8] proposed an approach for classifying malware and benign files based on both static and dynamic features. The static features used were function length frequency and printable string information. Dynamic features used are API function names and API parameters. The experiment was conducted on 2939 executable files including 541 benign files separately for every feature and then for the integrated method for Meta classifiers SVM, IB1, DT, and RF. The result showed that all Meta classifiers acquired the highest accuracy for integrated features and meta-RF is the best performer for all cases. They also compared their approach with other existing methods and found that their approach is showing better results.

Santos et al. [9] proposed an approach where both static and dynamic features were collected to train a malware classifier called OPEM. They used frequency of occurrence of operational codes, static features with execution trace of an executable, system calls, dynamic features to train the model to show that hybrid approaches perform better than both approaches run separately. They verified their approach using two different datasets along with Decision tree, KNN, Bayesian Network, and SVM.

D. VISUALIZATION APPROACHES

Malware analysis and classification can also be performed by using a visual representation of malware, thus making it an image processing problem. The malware images belonging to the same malware family are visually similar, and they differ noticeably from images belonging to other families. Thus with image similarity, the malware can be detected and classified using image recognition methods.

Nataraj et al. [10] were the first to explore the use of byte plot visualization for automatic malware classification. The malware samples were converted into byte plot representation and texture-based features were extracted for classification. They used GIST for computing the texture features of malware images. Their dataset consisted of 9,458 malware samples belonging to 25 different malware families which were collected using Anubis virtual environment. They used a global feature to train the KNN model with Euclidean distance as the distance measure to classify malware samples into respective classes. The results obtained showed that the image-based malware classification was much quicker than the existing dynamic approaches. The drawback of this approach was there was a huge computational overhead of computing the texture-based features.

Makandar et al. [11] proposed a malware classification and analysis technique was malware binaries are converted into grayscale images and are classified based on their texture features. They extracted texture-based global features using Gabor filter and GIST. The dataset used was Mahenhur for the experiment which consisted of 3131 malware binary samples from 24 different malware families. The texture features extracted were trained using an Artificial Neural Network for classification.

Aziz et al. [12] proposed the detection of malware classification model build using an image processing technique. Here image similarity approach is used to detect and retrieve viruses in the form of malware. They performed experimental result analysis on the Malimg dataset for experiment and showed that image processing techniques such as Normalization of malware grayscale images then applying wavelet transform using Discrete

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Wavelet Transform at three-level decomposition with PCA. The dimensionality reduction is done on the normalized image such as pre-processed malware. After decomposition, they applied wavelet-based statistical features such as mean, RMS, Standard Deviation, and Variance. The model gave an accuracy of 92.92% on the Maheneur dataset and 88.75% accuracy on the Malimg dataset.

Han et al. [13] proposed a malware detection and classification method using the op-code sequence to classify malware. The malware samples were converted into image matrices to visually represent malware which assisted in detecting features of malware and also in finding similarities between different samples swiftly. At first, the binary samples were disassembled using IDAPro and then opcode sequences were divided into blocks. They used two hash functions for each block of the opcode sequence and computed a coordinate and RGB value using the two hash function. The RGB values were plotted to the corresponding coordinates in the matrix of 8 by 8 dimensions to get an image matrix. To calculate the similarities between image matrices, they used selective area matching and evaluated their model on malware samples from 10 different families. Their results showed that image matrices of malware from the same family had a higher similarity score than with malware from different families. This shows that image matrices can effectively classify malware families.

Liu et al. [14] proposed a malware classification approach to automatically assign malware to their respective families and detect new malware. They used Opcode, import functions and grayscale byte plot as features. These features were then used for classifying the malware to their malware families and to identify new malware. To detect new malware the families used the Shared Nearest Neighbour clustering algorithm. The model was evaluated on a dataset consisting of 21,740 malware samples of 9 different malware families and reported an accuracy of 94.9% and detection accuracy of 86.7%.

Cui et al. [15] proposed a malware detection method based on deep learning. They converted malware samples to grayscale images. To address the data imbalance problem in the dataset bat algorithm was used. The malware images were classified using Convolutional Neural Network (CNN). The dataset consisted of 9,342 grayscale images of 25 malware families. They reported an accuracy of 94.5%.

III. PROPOSED SYSTEM

This section presents the proposed malware classification model using improved CNN, which includes: 1) Conversion of malware codes into RGB images. 2) Design for the RGB image classification model. Firstly, the malware binary files are converted into RGB images. Then convolution neural network is used to identify and classify the images. Based on the image classification results, the automatic recognition and classification of the malware software are realized.

A. VISUALIZATION OF MALWARE AS IMAGE

New malware is usually created by changing a small part of the existing malware. Thus by representing a malware as images helps in tracking such small changes. Inspired by this and the previous works, malware can be converted into RGB images. The steps involved in this conversion are as follow:

1. The malware raw binary file streams are converted into hexadecimal strings(0-15).
2. These hexadecimal strings are then segmented into 8-bit vectors which are also considered as an unsigned integer (0-255)
3. This one-dimensional vector is then transformed into 2D matrixes.
4. Every 8-bit vector of this array is then mapped into 256 shades of red, green, and blue colours.

(a) BrowseFox Family (b) Injector Family

Fig. 1. Illustration of Malware Images

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Fig.1 shows examples of malware images of various families. It can be noted that images of the same malware family are visually similar and they differ from the images belonging to other malware families. Fig.1 (a) shows the variants of the BrowseFox malware family, Fig.1 (b) represents malware variants from Expiro family. These visual similarities among the malware images of the same family have inspired to classify malware using image recognition methods. Here Convolution Neural Network with an additional pooling layer called Spatial Pyramid Pooling (SPP) is used for recognizing the malware images.

B. DATA PRE-PROCESSING AND AUGMENTATION

In this model, the data pre-processing step is nearly negligible as the input images need not be resized to a fixed dimension. So the pre-processing step includes the image to array conversion.

Data augmentation of the existing dataset of malware images is done to increase the diversity of data available for training the model. It is done to enlarge the dataset so that the neural network is exposed to different variations of the image. This can improve the robustness of the model. The techniques used for augmenting the data are rotation, flip, zoom, shift, etc.

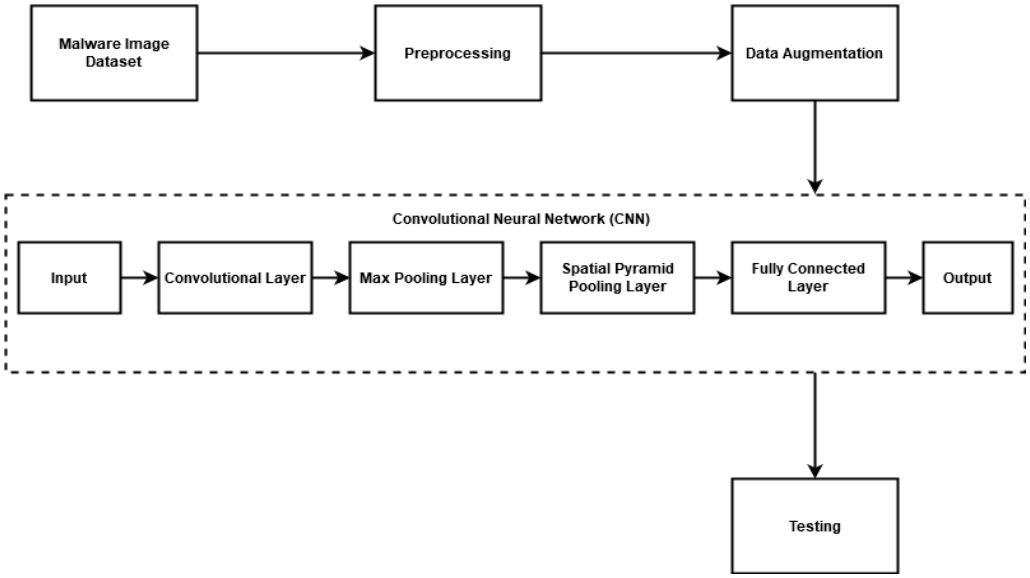

Fig.2. Proposed System

C. TRAINING

The network is trained using the malware images for classification. Here the training is done with Epoch as 4. The dataset is split into as training and validation set where 80% of the dataset is given for training the model and 20% of the dataset is kept for evaluating the model. Fig.2 represents the proposed system. This network accepts variable size images and generates fixed output due to an additional pooling layer called the Spatial Pyramid Pooling layer (SPP), hence eliminating the fixed dimension constraint.

D. CONVOLUTIONAL NEURAL NETWORK

The Convolutional Neural Network is an artificial neural network that is used for image recognition, classification, and processing. It uses deep learning to perform descriptive and generative tasks. Input to the model is a malware image and output is the class to which the malware image belongs. The structure of the CNN for malware color image is shown in the figure. The CNN used here consists of an input layer, convolution layer, max-pooling layer, spatial pyramid pooling layer, and fully connected layers. The activation function used with convolutional layers is ReLu (Rectified Linear Unit) and for the dense layer is softmax function.

In the pre-processing step, the malware color images are converted into the array and then given into the input layer of the neural network. Here the image size is variable, the batch size is 32 and the initial learning rate is le-3. First, the training images are brought into the neural network through the input layer. Following the input layer is the convolutional layer. The convolutional layer applies a certain number of convolutional operation to extract certain

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

features from the image. The filters operate on sub-regions of an image and feature maps are generated. The filter used in this system is a 3X3 filter. The activation function used in this neural network with the convolutional layer is ReLu (Rectified Linear Unit). ReLu is used to introduce nonlinearity after a linear operation is carried out. This layer increases the non-linearity property of the model without affecting the receptive field of a convolutional layer. The next layer after the convolution layer and ReLu is the max-pooling layer. Max Pooling layer helps in reducing the size of the input representation. Its function is to reduce the amount of parameter and computation. It extracts features from the feature maps. The result of max-pooling is a pooled feature map that highlights the most present features in the feature maps. It eliminates that information from the image that is not the feature and reduces the size and parameters. The pool size used is 2x2, so the max operation would be taking a max over 4 numbers in this case and the depth dimension remains unchanged.

Between the last convolutional layer and the dense layer a spatial pyramid pooling layer (SPP) is used. This layer generates fixed-length representation regardless of the image size. It pools the features and generates a fixed-length output, which is then given to the fully connected layer. The fully connected layer is a multilayer perceptron and uses a classifier at the output layer. Every neuron from the previous layers is connected to the neurons in the next layer. This layer uses the features from the output of the previous layers to classify the input image based on training data.

E. TESTING

After training the model, testing, or evaluation of the model is done to determine the accuracy and precision of the model. It determines how accurately the model can classify a malware image and results in its corresponding malware family.

IV. EXPERIMENTAL SETUP

The dataset consists of 3500 malware images from 10 malware families. Each of the malware images belonged to one of the 10 malware families. The malware samples were randomly split into a training set and testing set. 80% of the malware sample was kept for training and the remaining 20% of malware samples were for testing evaluating the model. For implementing the proposed model we used Python programming with different python libraries. For learning the model, Keras and Tensorflow backend was used. Scikit-learn was used for performance evaluation and matplotlib was used for plotting. We ran our experiment on PC with Intel Core i5, CPU (1.6 GHz), and 4GB RAM.

V. RESULT ANALYSIS

We implemented the malware classification model using RGB images with CNN architecture and RGB images with the CNN+SPP model. The CNN model was trained and tested using the image of size 224X224. The proposed CNN+SPP model trained variable-sized malware images of sizes. Fig.3 represents the performance of the two models on the training dataset. It can be referred that the accuracy of the training dataset for the proposed model is higher than the CNN model. Fig.4 represents the performance of the classification models on the validation dataset. It is clearly shown that there is an increase in the accuracy during each epoch value in the proposed system.

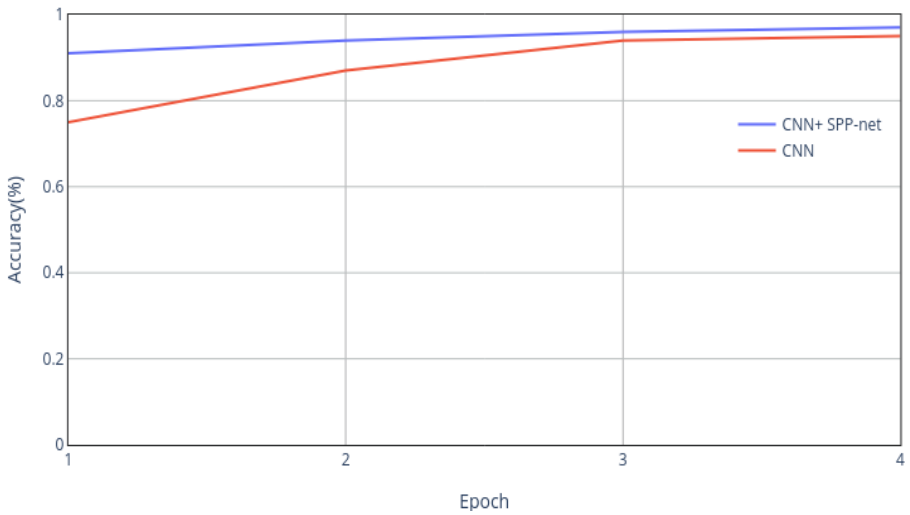

Fig.3. Accuracy of classification models on the Training set

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

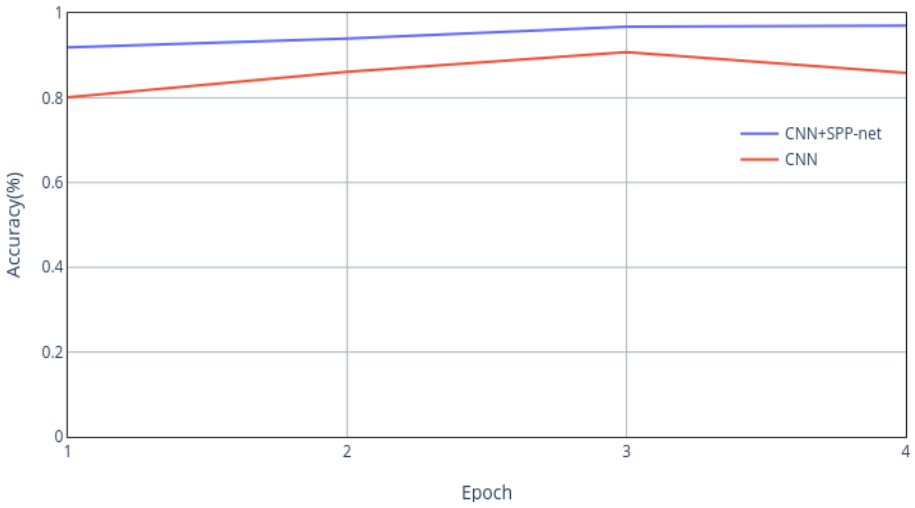

Fig.4. Accuracy of classification models on the Validation set

The dataset consists of 3500 images in which 2800 images were used for training the model and 700 images were used for evaluating the model. The proposed model gives 97% of classification accuracy which is higher than the basic CNN model which gives an accuracy of 85%.

VI. CONCLUSION

Malware has become a great threat to the security of computer systems. It is essential to prevent the attacks and impact that they cause to the system. In this paper, we presented an approach for malware classification based on the visualization approach. Malware images are represented as RGB images. The model proposed in this work gives an improvement in previous works and shows that the SPP layer can improve all CNN based networks as it increases recognition accuracy and improves the robustness of the model.

The malware authors use obfuscation techniques such as adding jumping instructions, redundant code fragments to make malware remain undetected from the visualization approaches. This can be an area for future work where implementation can also be done for obfuscated malware.

REFERENCES

[1] Z. Zhao, "A virus detection scheme based on features of the Control Flow Graph. ” 2nd International Conference on Artificial Intelligence, Management Science, and Electronic Commerce (AIMSEC), pages 943-947, 2011.

[2] Y. Ye, D. Wang, T. Li, and D. Ye, “Imds: intelligent malware detection system, ” in KDD, P. Berkhin, R. Caruana, and X. Wu, Eds., pp. 1043-1047, ACM 2007.

[3] Zolkipli, M.F. and Jantan, A. (2011) An Approach for Malware Behavior Identification and Classification. Proceeding of 3rd International Conference on Computer Research and Development, Shanghai, 11-13 March 2011, 191-194.

[4] C. Fan, H.W. Hsiao, C.H. Chou, and Y.F. Tseng, "Malware Detection System Based on API Log Data Mining". IEEE 39th Annual International Computers, Software Applications Conference, 2015.

[5] S.Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with a deep neural network using process behavior, " in Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual, vol. 2. IEEE, pp. 577 – 582, 2016.

[6] Biley, M., Oberheid, J., Andersen, J., Morley Mao, Z., Jahanian, F. and Nazario, J., “Automated Classification and Analysis of Internet Malware”. Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection, 4637, 178-197, 2007.

[7] Anderson, B., Storlie, C. and Lane, T., Improving Malware Classification: Bridging the Static/Dynamic Gap. Proceedings of 5th ACM Workshop on Security and Artificial Intelligence (AISec), 3-14,2012.

[8] Islam, R., Tian, R., Battenb, L. and Versteeg, S., Classification of Malware Based on Integrated Static and Dynamic Features. Journal of Network and Computer Application, 36, 646-556, 2013.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

[9] Santos, I., Devesa, J., Brezo, F., Nieves, J., and Bringas, P. G. "OPEM: A Static-Dynamic Approach for Machine-Learning-Based Malware Detection",International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions pp 271-280, 2012.

[10] L. Nataraj, S. Karthikeyan, G. Jacob, and B. Manjunath. "Malware images: visualization and automatic classification". In Proceedings of the 8th international symposium on visualization for cyber security page 4. ACM, 2011.

[11] Makandar, A., Patrot, A, Malware analysis and classification using artificial neural network. In: 2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15), IEEE, pp. 1–6, 2015.

[12] A. Makandar and A. Patrot, "Detection and retrieval of malware using classification," International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017.

[13] K. Han, J. H. Lim, and E. G. Im, "Malware analysis method using visualization of binary files". In Proceedings of the 2013 Research in Adaptive and Convergent Systems, pages 317– 321. ACM, 2013.

[14] Liu, L., Wang, B.S., Yu, B., Zhong, Q.X.: \textit{Automatic malware classification and new malware detection using machine learning}. Front. Inf. Technol. Electron. Eng. 18(9), 1336–1347, 2017.

[15] Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G., & Chen, J. (2018). "Detection of Malicious Code Variants Based on Deep Learning". IEEE Transactions on Industrial Informatics, 14(7), 3187-3196, 2018.