Malware Classification using Neural Network

Ashwini Sapkal
Dept. of Information Technology
Army Institute of Technology
Pune, India
asapkal@aitpune.edu.in

Deeptanshu Singh Rathore
Dept. of Information Technology
Army Institute of Technology
Pune, India
deeptanshurathore28@gmail.com

***Abstract*—The explosive growth of malware variants poses a major threat to information security. Traditional anti-virus systems based on signatures fail to classify unknown malware into their corresponding families and to detect new kinds of malware programs.In this paper a comparative analysis is done on two approaches that can be used for this classification on a dataset provided by Microsoft in ”Microsoft Malware Classification Challenge (BIG 2015)”. First approach is using grey scale images that are formed by reading the assembly file of malwares as binary file and converting that binary file to grey scale image . We apply a CNN model on these images for classifying them in different classes of malwares . Second approach is based to text classification using the assembly files of malwares and extracting features like count vector and n-gram.We built a Deep Neural Network model on these features for classification .**

***Index Terms*—Malware classification ; Machine Learning ; Deep Learning ; n-gram ; count vector ; Convolutional Neural Network ; grey-scale image**

I. Introduction

Internet has become an integral part of everyone’s daily life.As recorded in a survey of 2017 around 57% of world population is now connected over Internet. Internet is used for different commercial and non-commercial activities like banking , communication , entertainment , shopping.Albeit making everyone’s life convenient , Internet has exposed us to the possibility of getting attacked by different means and sources.Illegitimate users use malware programs to commit financial frauds or steal private and sensitive information from legitimate users.

Malware are the malicious softwares that cause damage to users data , computers or networks in some or the other way [1].Malware contains malicious programs like worms , backdoors , Trojan horses .Currently, malware is an important challenge in the field of information security. According to a report from Kaspersky Labs (2015), in the past year, 58% of corporate computers were attacked and 29% of companies suffered from network attacks.Although there are hundreds and thousands of new malware found every day, but most of them are derived from the known families of malware and only differ very little from these well known families.

Most of the malware detection systems and anti-virus software use signatures and anomaly detection methods for classifying a file as malware or a genuine file. These signature based methods have great accuracy for the well known malwares that have been previously encountered by the system , but fails in detecting new type of malwares [2].This technology is mainly dependent on the experience of experts , and needs to spend a lot of time and funds in constantly updating the feature library [3]. New malwares can be found using anomaly detection method but the false alarm rate is very high.

Malware analysis is being used both for detecting malware and classification of malware. Malware detection is the process of labelling a binary executable as benign or malign. So basically, it is the first step of malware analysis. Only after we know a file is a malware we proceed for further analysis. Whereas classification is the process of figuring out the class to which a given malware belong.Based on the state of the analyzed malware , malware analysis can be done using two methods , static analysis and dynamic analysis.Static analysis refers to the analysis of the program without executing it. The major advantage of static analysis is can easily find author’s style and profile the code flow but there is a disadvantage also that is , it is easily opposed by obfuscation techniques.Dynamic analysis involves executing a binary and observing its behaviour to deduce whether it is a malware or not. The malware is executed in a sand-boxed environment, or VM, so that the effects of the malware can be contained.However , this method spends time not only on debugging the program , but also on tracking the running process of the program.Therefore , dynamic analysis is usually in inefficient as compared with static analysis.

With the successful application of Machine Learning mainly Deep Learning in the fields of image processing, speech recognition and other applications [4]. In past few years Machine Learning has become an important part of malware detection and analysis in the information society. Malware classification using Machine Learning mainly constitute of three steps feature extraction, building a Machine learning model and classifying malwares into different classes.

In the paper a experimental comparative analysis of two different methods is done that can be used for malware classification and detection based on Neural Network model.(1) Converting the assembly files of malwares into grey scale images by reading them as binary files and making pixels using the binary codes and they building a CNN model on these images for classification. (2) Performing text classification using same assembly files and extracting features like count vector and ngrams and supplying these features to a multi layer neural network for classification.Then finally performed a comparative analysis on both the techniques used .

II. RELATED WORK

In [5] the authors visualizes malware into grey scale images.They used GIST descriptor on the malware images. The GIST descriptor is mostly used for scene classification.They have used K-nearest neighbour for classification.

In [6], Aziz Makandar and Anita Patrot applied Discrete Wavelet Transformation (DWT) on the malware images to extract features. They use Support Vector Machine (SVM) to discriminating the malware classes.

Gonzalez et al. [7] gave a different approach for malware classification baseed on features extracted by counting the times of each library in the malware.For authentication their method, they collected a large number of malwares from Kaspersky, F-Secure, Norton and McAfee . The implementation used Euclidean Distance and Multilayer Perceptron.

Lin et al. [8] proposed a method for malware classification which uses SVMs to classify malicious softwares.The high-dimensional aspects of th n-gram features were constructed by dynamic analysis . They gave a two-stage dimentionnality reduction method.First, they applied term frequency-inverse document frequency(TF-IDF) to select important features. They used principal component analysis(PCA) to reduce the number of random variables . Multiple groups of experiments showed the effectiveness and efficiency of their method.

Kapoor and Dhavale [9] used somewhat similar approch to extract n-gram features . Bi-normal separation is used for feature scaling. They used information gain for the selection of relevant features.Through this they achieved an accuracy of 0.972 for multiclass classification.

In [10] the authors used gray-scale images, Opcode n-gram, and import functions for extracting features.They used shared nearest neighbor (SNN) clustering algorithm in their approach.The results show that their system can effectively classify the unknown malware with a best accuracy of 98.9%, and successfully detects 86.7% of the new malware

III. OUR METHOD

A. System Overview

Fig. 1 is the flow diagram of the system containing all the steps involved in the experiment.Step includes feature extraction , model building , classification and analysis of results.

1) Feature Extraction: In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is a dimensionality reduction process, where an initial set of raw variables is reduced to more manageable groups (features) for processing, while still accurately and completely The performed experiment considers two types of features for textual classification,count vector and trigrams.

2) Model Building: A machine learning model can be a mathematical representation of a real-world process.The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from.The term ML model refers to the model artifact that is created by the training process.For the classification based on images a 5 layer Convolutional Neural Network model is used and for the textual classification a 5 layer neural network model is used.

3) Analysis of result: A same set of samples are used for training and testing in all the approaches and the results are compared based on the accuracy score.

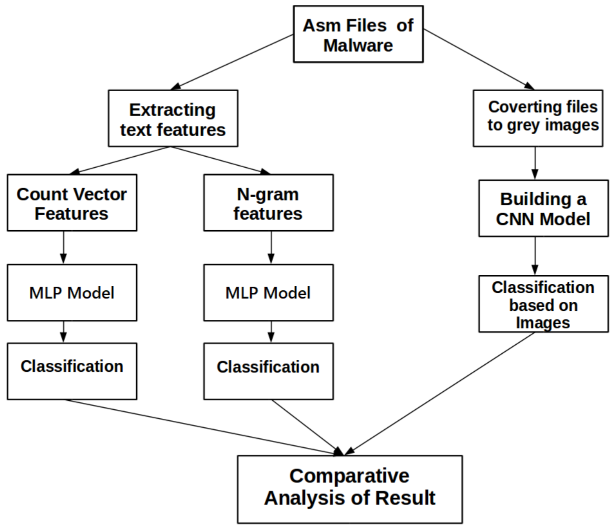


Fig. 1. System Overview

B. Grey-Scale Images

Malware classification based on gray-scale imaging is a novel approach. It has been proven as an effective static analysis tool. A gray-scale image is an image that is expressed in gray scale. The brightness of white to black is divided into 256 grades according to a logarithmic relationship.Fig. 2 shows converted image of a assembly file.

To convert the assembly file to grey scale image , first the assembly file is read as a binary file so that the file only contain values 0 and 1.Then the file is read as a stream of bits and make a group of 8 bits so as it can be converted to an unsigned integer in the range of 0255. In a gray-scale image, 0 and 255 represent black and white, respectively. Finally, the transformed file is mapped to a matrix called the gray-scale matrix. Now as the size of different files is different so will the images will be. So to standardize them we consider that the width of the image will be fixed and the length can vary, so that it can be fed into the neural network. Fig. 3 shows the process of converting the assembly file to its coreesponding grey-scale image.

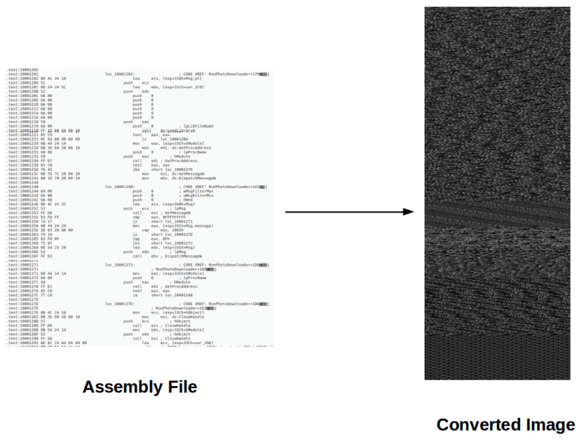


Fig. 2. Example of converted Image

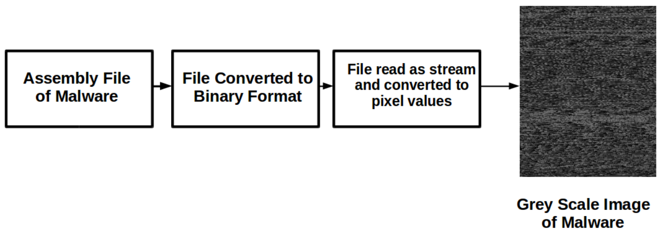

Assembly File of Malware
File Converted to Binary Format
File read as stream and converted to pixel values
Grey Scale Image of Malware

Fig. 3. Process of converting assembly file to grey-scale image

C. Textual Feature Extraction

There are many different techniques used to extract different features from textual data like Word Counts , Word Frequency , Hashing and n-grams. In our approach we compare the accuracy by using two features Count vector and opcode 3-grams.

1) Count Vector: The count vector feature extraction method tokenizes a text document and build a vocabulary of known words, and encode new documents using that vocabulary. Count Vector transforms the data and create a vector with a length of the entire vocabulary and an integer count for the number of times each word appeared in the document.Now this transformed vector is used as a feature for building a model and later classifying the documents into their corresponding classes.In this experiment a dictionary of top 300 most commonly occurring words is built from all the files and transformed the text according to it.

2) n-gram features: n-gram is an effective method for text feature extraction [11].It is based on the hypothesis that the appearance of the n words is related only to the previous n1 words, wherein n represents the length of a characteristic sequence.The model finds feature sequences in a sliding window way.It creates a dictionary for most common ngrams.In the experiment top 300 trigrams are used and the files are transformed according to this dictionary.

D. Multi Layer Neural Network Model for Text Classification

Deep Learning is a sub branch of Machine Learning that became popular in recent times.It is basically a multi layer neural network with more than one hidden layer.For this experiment a multi layer neural network model with an input and an output layer along with three hidden layer is used.In the hidden layer relu is used as the nonlinear activation function and at the output layer softmax activation is used to get the probability distribution among different classes. For both the count vector feature as well as for trigram feature the same structure is used for building the model so as to compare the accuracy of both the approaches. Fig. 4 shows a structure of a deep neural network with one input and one output layer along with 3 fully connected or dense hidden layers.

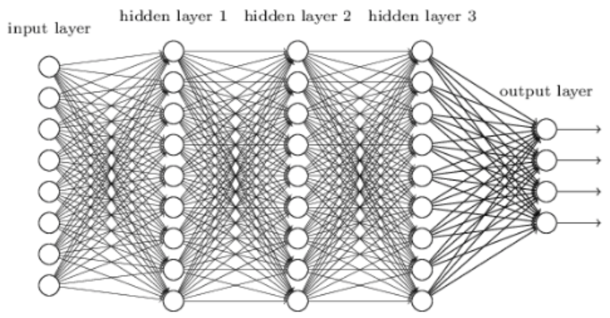


Fig. 4. Architecture of used Deep Learning Model

E. CNN Model for Image Classification

In recent times Convolutional Neural Networks have proven to perform better for the task of image classification. The idea behind CNN model is that a local understanding of an image is good enough.The fewer parameters greatly improves the time it takes to learn as well as reduces the amount of data required to train the model.Instead of using a fully connected network of weights from each pixel , CNN has enough weights to look at a small patch of the image.Any CNN model consists of 3 types of layers : (1) Convolution Layer - Convolutional layers apply a convolution operation to the input, passing the result to the next layer.(2)Pooling Layer - It combines the outputs of neuron clusters at one layer into a single neuron in the next layer.(3)Fully Connected Layer - Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

In the experiment CNN structure with an input layer followed by 2 pairs of convolution and max-pooling layer, and then finally a fully connected layer for image classification is used. Since we have used a small subset of the whole dataset, we have augmented the converted images by using flipping, scaling and rotating them. Fig. 5 shows a structure of a CNN Model with one input and one fully connected layer along with 2 convolution and max-pooling layer.

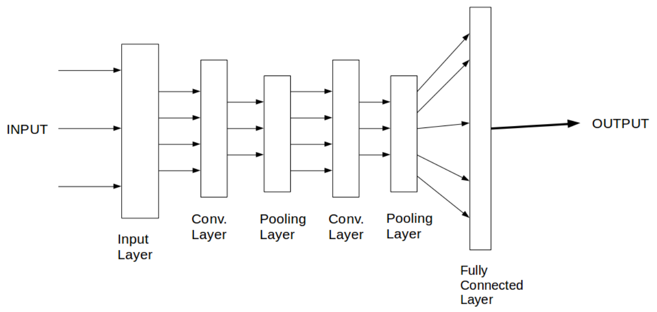


Fig. 5. Architecture of used CNN Model

IV. EXPERIMENT

A. Dataset

For this experiment dataset provided by Microsoft is used which was used for " Microsoft Malware Classification Challenge (BIG 2015) " [12].The Dataset contains 21,741 malware files representing a mix of 9 different families which are Ramnit, Lollipop, Kelihosver3, Vundo, Simda, Tracur, Kelihosver1, ObfuscatorACY, Gatak. Since the dataset is huge and we don't have such big computational resources, for the experiment a small subset of the dataset which contains 10 files belonging to each 9 classes, total 90 samples is used.Malware file has varying size from 1Mb to 32Mb.

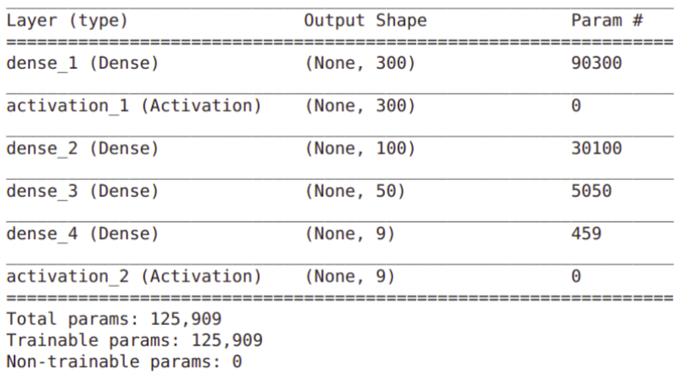
B. Classification

1) Based on Grey-Scale Images: The converted image dataset was given a train-test split of 0.2 i.e. from the 90 samples 72 were used for training and 18 for testing.The training images were augmented to increase the size of training dataset. The CNN model with one input and one fully connected layer along with 2 layers of convolution and max pooling was trained. The model used cross-entropy as the loss function , adam optimizer and the accuracy matrix for result generation.

2) Using Count vector as feature: Implementation was done by creating a dictionary using all the 72 training files and taking top 300 most occurring words. Then all the files were transformed according to this dictionary and these files with count vector features were feeded to a deep neural network created using keras python library for training.The structure of neural net used consists of 5 layers 1 input layer , 1 output layer and 3 hidden layer.

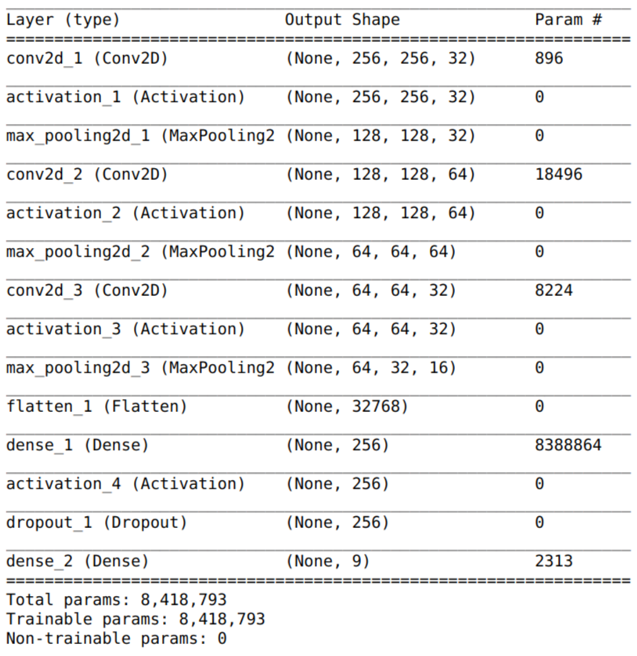
3) Using trigrams as feature: The implementation was done by creating trigrams of all the files using nltk python library. Then a dictionary was built of top 300 trigrams taken by reading all the files . The complete training files and testing files were transformed according to this dictionary. For classification same structure of Neural Network was used as that for count vector.

4) Model Structure and Parameters: The model used cross-entropy as the loss function, adam optimizer as the optimizing alogithm and the accuracy matrix for result generation.Given below is the summary of model used.Input layer contains 300 neurons as the size of input vector(dictionary) is 300, the first layer contains 100 neurons and relu as the activation function, the second hidden layer have 50 neurons and the final fully connected output layer have 9 neurons as the number of classes given are 9 and softmax is used as the activation function as we want a probability distribution among the classes. The CNN model used for the experiment took 256*256 images as the input with a filter size of 3*3 in first layer.Subsequent layers used a filter of 3*3 and 2*2 size. The pool size used in the experiment is 2 and the learning rate of 0.004 is used.The dense layer has 256 neurons and a dropout of 0.5 is used before the fully connected layer to normalize the model.



```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 300)               90300
_________________________________________________________________
activation_1 (Activation)    (None, 300)               0
_________________________________________________________________
dense_2 (Dense)              (None, 100)               30100
_________________________________________________________________
dense_3 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_4 (Dense)              (None, 9)                 459
_________________________________________________________________
activation_2 (Activation)    (None, 9)                 0
=================================================================
Total params: 125,909
Trainable params: 125,909
Non-trainable params: 0
```

Fig. 6. Model Summary of MLP Model



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 256, 256, 32)      896
_________________________________________________________________
activation_1 (Activation)    (None, 256, 256, 32)      0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 128, 128, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 128, 128, 64)      18496
_________________________________________________________________
activation_2 (Activation)    (None, 128, 128, 64)      0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 64, 64, 32)        8224
_________________________________________________________________
activation_3 (Activation)    (None, 64, 64, 32)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 64, 32, 16)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 32768)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               8388864
_________________________________________________________________
activation_4 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 9)                 2313
=================================================================
Total params: 8,418,793
Trainable params: 8,418,793
Non-trainable params: 0
```

Fig. 7. Model Summary of CNN Model

C. Result and Comparison

For exact comparison the same dataset for all the 3 models is used and a train-test split of 0.2 for all.The criteria used for comparison is the accuracy score.

- The CNN model built for classification based on the grey-scale images gave an accuracy of 81.7%.
- The Deep Neural Network with count Vector as feature gave an accuracy of 80.6%.
- The Deep Neural Network with count Vector as feature gave an accuracy of 87.03%.

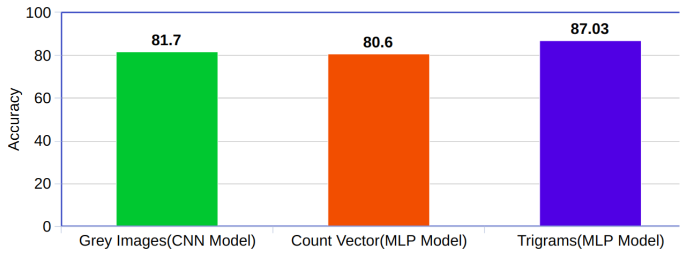


Fig. 8. Experimental Result Comparison

V. CONCLUSION

From the experiment we have found that converting assembly files to images and applying classification on them takes less computation time as compared with doing textual classification.But the trigrams shows better accuracy in all as the instructions in assembly code are combination of 3 words. Since a subset of the complete dataset is taken, accuracy is low but when the models will be trained with the complete dataset the accuracy will me much higher than what observed.Other factor that can increase the accuracy is the size of dictionary what we have used in both the cases,count vector and trigrams.

For further analysis there are various pre trained model for image and text classification, we will be fitting are dataset onto these models and compare the results given by these models.

REFERENCES

[1] Malware definition, https://techterms.com/definition/malware, 2017, accessed: 2017-03-14.

[2] Lee J., Jeong K., Lee H., 2010. Detecting metamorphic malwares using code graphs. Proceedings of the 2010 ACM symposium on applied computing. ACM, p. 1970-1977.

[3] Damodaran A., Di Troia F., Visaggio C.A., et al., 2015. A comparison of static, dynamic, and hybrid analysis for malware detection. Journal of Computer Virology and Hacking Techniques, p. 1-12.

[4] L.P. Wang and X.J. Fu, Data Mining with Computational Intelligence, Springer, Berlin, 2005..

[5] Nataraj L., Karthikeyan S., Jacob G., Manjunath B. S.,The malware Images: Visualization and Automatic Classification,International Symposium on Visualization for Cyber Security (VizSec) ,July 20, 2011,Pittsburg, PA, USA

[6] Aziz Makandar and Anita Patrot, Wavelet Statistical Feature Based Malware Class Recognition and Classification using Supervised Learning Classifier, Oriental Journal of Computer Science and Technology, ISSN: 0974-6471, June 2017, Vol. 10, No. (2): Pgs. 400-406

[7] Gonzalez L.E., Vazquez R.A., 2013. Malware classification using Euclidean distance and artificial neural networks. Artificial Intelligence (MICAI), 2013 12th Mexican International Conference on. IEEE, p.103-108.

[8] Lin, C.T., Wang, N.J., Xiao, H., et al., 2015. Feature selection and extraction for malware classification. J. Inform. Sci. Eng., 31(3):965-992. https://doi.org/10.6688/JISE.2015.31.3.11

[9] Kapoor, A., Dhavale, S., 2016. Control flow graph based multiclass malware detection using bi-normal separation. Defen. Sci. J., 66(2):138-145.https://doi.org/10.14429/dsj.66.9701

[10] Liu LIU,Bao-sheng WANG,Bo YU, Qiu-xi ZHONG , ”Automatic malware classification and new malware detection using machine learning ”, Front Inform Technol Electron Eng 2017 18(9):1336-1347

[11] Jain S, Meena Y K. Byte level n-gram analysis for malware detection. Computer networks and Intelligent Computing. Springer Berlin Heidelberg, 2011:51-59.

[12] Dataset , ” Microsoft Malware Classification Challenge (BIG 2015) ’ ’ ,arXiv:1802.10135