Malware Detection Methods & Comparative Analysis using Machine learning & Deep learning Techniques

Arpit Saxena*, Apratihat Singh**, Sagarika Sardesai***
*(Computer Science with specialization in Information Security, VIT University, Vellore
Email: arpitsaxena0910@gmail.com)
** (Computer Science with specialization in Information Security, VIT University, Vellore
Email: apratihat11@gmail.com)
***(Computer Science, VIT University, Vellore
Email: sagarikasardesai13@gmail.com)

--************************----------------------------------

Abstract:

There is always a constant increase in the number of Malwares, which belong to different families and are designed to fulfill separate purposes of creation, and thus to avoid the toll of damage caused by these Malwares, we need to understand their basic functionalities. The traditional method to detect Malwares used by various anti-viruses, is a signature based method, which identifies malwares based on a local signature database, but this technique is being rapidly exploited by newly devised malwares that use evading techniques like encrypting, packing etc to avoid detection. Due to this, new technologies in the field of Machine learning and deep learning as well as its combination with other fields are being used to cluster different approaches in order to accurately detect these Malwares and to understand their functionalities.

Keywords — **Malware Detection, Malware Visualization, CNN, Ensemble Method, PE Headers, Malware Classification**

--************************----------------------------------

I. INTRODUCTION

Malwares as the name suggests are malicious softwares, that intend to harm computer systems by damaging and disrupting the existing software or network, moreover they can be easily used by an attacker to gain unauthorized access to the system. Malwares can be of different types and thus can attack the system in a variety of ways. Hence, in order to safeguard the system against potential attacks by malwares, we need to determine the type of malware used in the attack and its various drawbacks. This is where Malware Analysis comes into picture, Malware Analysis is basically, an information gathering process of a particular Malware. Malware Analysis, gives an in-depth detail about a Malware, ranging from the type of Malware it is, to, its objective. Malware Analysis can be subdivided into two broad categories-

i) Static Analysis: A safer way to analyze a Malware compared to its counterpart, static analysis can be used for rapid detection of a large number of Malwares, by using statistical approaches to not only determine whether a file is a Malware or not but also to get information about its objective and the command it is executing. Being the easier approach among the two categories, it still has a prominent weakness, which is that static analysis can be easily deceived by the Malwares using techniques like Encryption and Obfuscation.

ii) Dynamic Analysis: Another approach to determine whether a file is Malware or not, dynamic analysis also gives detailed information regarding the behaviour of the file. This is executed by running the suspected file in a safe environment and according to the disruptions caused or malicious content released by this file, the family and the objectives of the Malware present in it is determined. Being a thorough analysis technique, it is highly time consuming as many malwares begin their actual attack after a long time.

Taking into account, both these analysis methods, it is evident that Dynamic Analysis gives a more detailed study about the Malware, but the fact that outshines its advantages is that in Dynamic Analysis the malicious file has to be run and tested, which can cause grave problems of its own. Hence, Static Analysis is the preferred approach. New approaches use Machine Learning along with static analysis to determine the maliciousness of a file. But following the increase in new techniques to evade Malware, new smarter Malwares are easily deceiving these techniques. Hence, we will also use Deep Learning in combination with other information such as PE Headers of a Malware file to determine whether a file is Malware or not, if yes, then the details about its family, objectives and ways to prevent it will be computed.

II. RELATED WORK

With the rise in the number of smart devices and users, malware of numerous types and increasing complexity are found every day. The malware creators can generate metamorphic and obfuscated malware with great ease, that can evade detection using inbuilt automated toolkits. Several methods have been recommended for the analysis of these malware, with the additional computational overhead that is quite significant in size. This results in the wrongful classification for malware data sets that are large in scale and are complex. [1] J.Fu proposes a method which is practically based on visualizing a given malware and determining its class both globally and locally, this was his main idea to ensure precise classification of malwares. The malware is visualised as RGB coloured images and their global features are extracted. Fu selected Gray-level matrix and color moments to describe the texture and color features respectively; this helped in reducing the complexity of the model as this technique produced low-dimensional feature data.

[2] L.Liu recommends the identification of these variants by the construction of controlled dissemble of the malware files. These dissembled files are then converted into grayscale images and local mean method compresses these images, to obtain increased efficiency. They are then mapped into feature vectors and classification of malware is done based on novel ensemble learning, based on means and diversity selection.

Majority of users prefer Android which has made Android devices susceptible to malware attacks. According to [3] F. M. Darus the Android malware will be visualised into gray scale images and their image features are extracted using GIST descriptor. The detection is carried out and compared using three different classifiers namely k-nearest neighbor (KNN), Random Forest (RF), and Decision Tree (DT).

Since the usage of obfuscation strategies for the evasion of malware is one of the biggest concerns concerning security, it needs to be addressed. [4] R.Kaur in their paper introduces a fingerprinting approach which is novel, for Android obfuscation tools based on spatial analysis. The first order and second order statistical features are to be analysed for further analysing the spatial distribution of pixels that represent the obtained binary images.This method has achieved 90% accuracy in fingerprinting numerous tools with certain configuration options.

III. DETECTION METHODOLOGIES

In this section we introduce different methodologies for Malware detection and classification. It involves two major steps:

1. Malware Detection: A binary classification method to identify whether a given raw file is malicious in nature or not.
2. Malware Family Classification: If the given file is found malicious, we then further identify which class or family the given malware belongs to.

A. Malware Detection and Family Classification Using Convolutional Neural Networks(CNN).

In this method we tend to use Convolutional Neural Networks(CNN) for malware detection and family classification. The detection and family classification process involves three stages. They are described as follows:

1. The very first stage involves the data preprocessing. Here our data consist of Windows executable files, which needs to be converted into grayscale images. After the conversion of all the raw files into grayscale images, this stage comes to an end and thus provides us with the image dataset, which will be used to train(provide input) the CNN model.
2. In the second step, the CNN model is then used to predict the nature of the testing data. It predicts whether the new file is a malware or not.
3. Finally if the given file is found malicious, we then identify which family does our malware belong to.

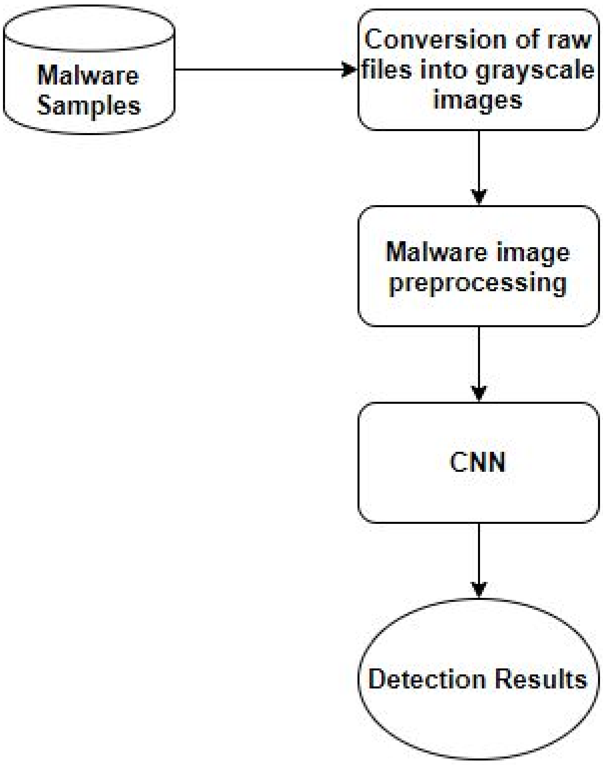

Malware Samples
Conversion of raw files into grayscale images
Malware image preprocessing
CNN
Detection Results

Fig. 1 Flow of Malware Detection Using CNN

1) Conversion of Raw files into Grayscale images: The very starting point of this method requires us to convert these raw files into a grayscale image which can be used to train CNN model. The executable file is viewed in binary format. This binary file is then read as a vector of 8 bit. The minimum and maximum value which the 8 bit vector can represent are 0(black) and 255(white) respectively, thus representing the grayscale pixel value. This pixel value is then stored in the 2X2 vector(matrix). This matrix can thus be visualized as a grayscale image. Hence we successfully converted a raw file into a grayscale image. The size of the malware images is dependent on the size of the raw file and therefore all images are of different sizes. Since CNN needs all the input images to be of the same size, we need to normalize all the malware images. For the normalization process, we tend to use bilinear interpolation algorithm as it uses four nearest pixel values of the original image to determine the new pixel value after normalization, and thus performs better than the nearest neighbor interpolation. It is also important to understand that larger the image size, richer data would be available for CNN to learn, and therefore we decided the normalized size of the grayscale images to be 256X256.

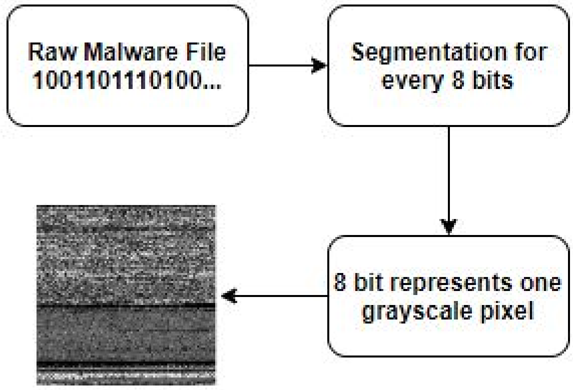

Raw Malware File
1001101110100...
Segmentation for every 8 bits
8 bit represents one grayscale pixel

Fig. 2 Conversion of raw file into grayscale images

2) Similarity between the Malware images belonging to the same family: After the conversion of raw files into grayscale images we also observed that the malware files belonging to the same family look very similar to each other. You can refer to Figure 3. All the images belong to the same malware family i.e. Ramnit. Hence, based on this observation we concluded that malware files belonging to the same family look alike when converted to grayscale image, thus we used CNN to determine the family of the malware.

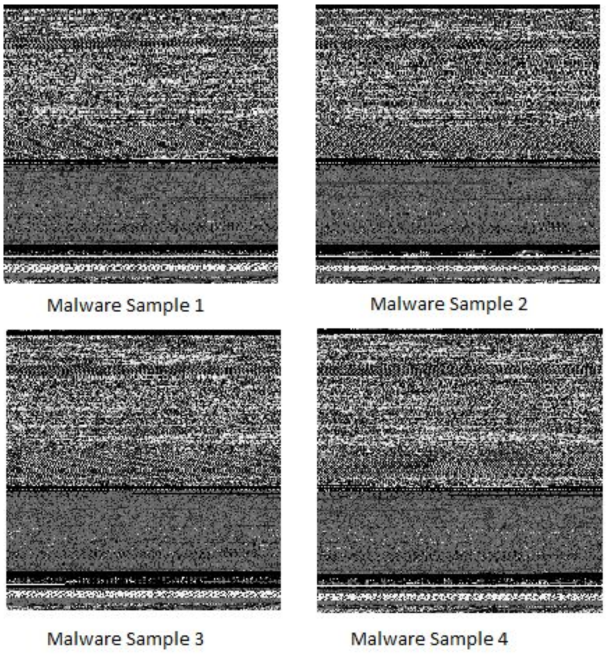


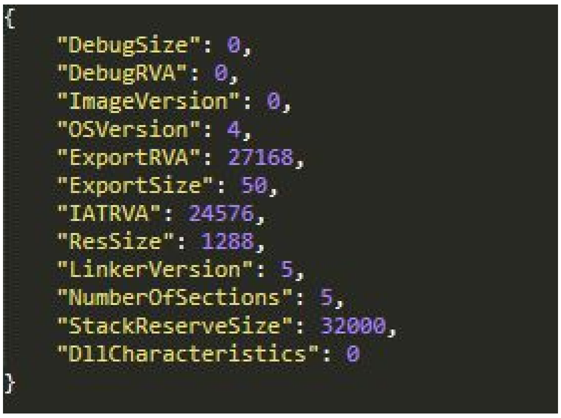
Fig. 3 Malware Samples belonging to family Ramnit

B. Malware Detection using PE Header

In this method we tend to use the Portable Executable header data of the raw files to create a Support Vector Machine(SVM) model to provide a binary classification about the nature of the raw files(malicious or not). PE header consists of meaningful data about the file such as Debug Size, Debug RVA(Relative Virtual Address), OS Version, Export RVA, Number of sections, DLL Characteristics etc. This process involves following four stages:

1. Extraction of the PE header data from each raw data file.
2. Save the extracted PE header data in a database or dataframe(for python).
3. Use the above collected data to train the SVM(Support Vector Machine) model.

You can refer to Figure 4 in order to understand how a PE header of a windows executable file looks like. This json data is then stored in the database which is then used by our SVM model to learn the features of Malware.



```
{
    "DebugSize": 0,
    "DebugRVA": 0,
    "ImageVersion": 0,
    "OSVersion": 4,
    "ExportRVA": 27168,
    "ExportSize": 50,
    "IATRVA": 24576,
    "ResSize": 1288,
    "LinkerVersion": 5,
    "NumberOfSections": 5,
    "StackReserveSize": 32000,
    "DllCharacteristics": 0
}
```

Fig 4. PE header file format

C. Malware Detection Using Ensemble Model

In this particular method, we integrate the above two methods discussed(A and B) by stacking ensemble. The above two models are weaker models when they make the prediction alone, and therefore this method helps us to build a stronger model. So in order to generate a final detection result, we use the outputs produced by the above two methods(A and B) as the input to our Logistic Regression model. A temporary result is generated by our two models(A and B), which are also referred as first level models. These temporary results are provided as input to our final model(Logistic Regression) which stacks the outputs of first level models to produce a better outcome. This model is also referred to as a second level model. You can refer to Figure 5 to understand the flow of this method.

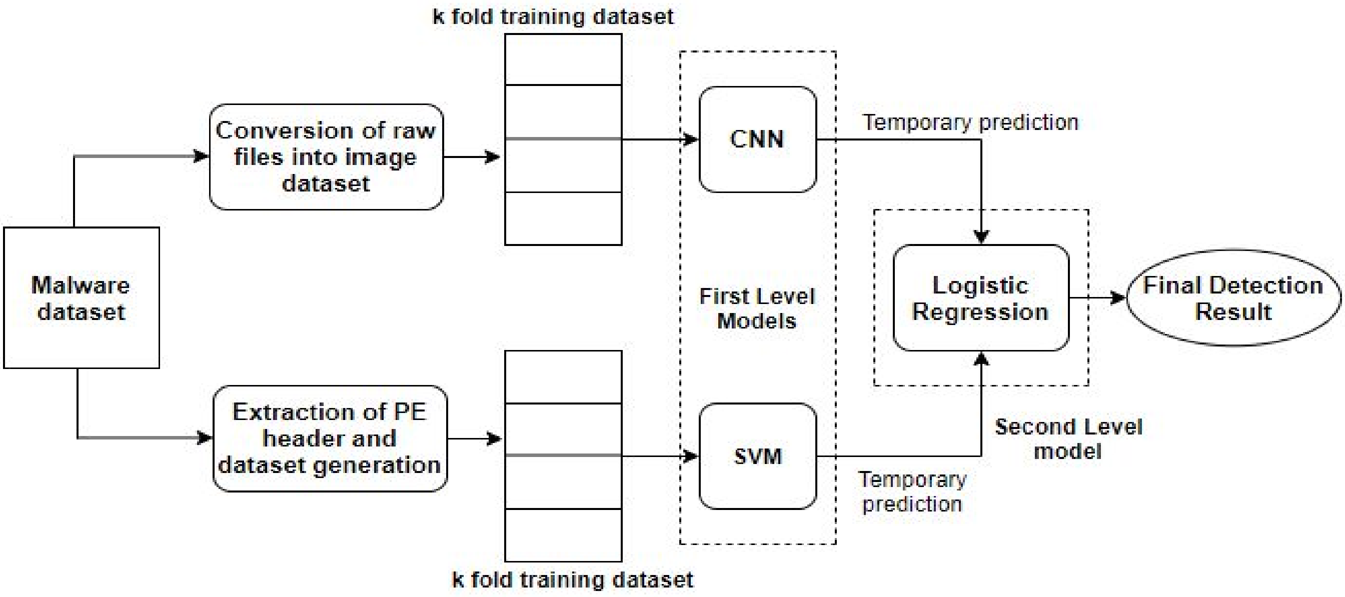


Fig 5. Diagrammatic view of Ensemble method

IV. RESULTS AND CONCLUSION

Hence with help of any of the following methods we can develop a machine learning model to predict whether the given file is malicious in nature or not. But in order to understand which of the following methods gives the best result, we need to look into the statistics of the predictions made by each model. The dataset used by our paper is a publicly available malware dataset from Microsoft released in 2015. The dataset is a mixture of malware files belonging to 9 different families. The family names are Ramnit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator.ACY, Gatak. The first method in which we used CNN(III.A) to train malware images provided us with some promising results. We achieved an accuracy of **96.49%**. The second method(III.B) which involved the prediction of malicious files used PE header as the dataset. The SVM model was good enough with it's prediction, but not as good as the first one. We achieved the accuracy of **90.22%** using this method. The final ensemble method(III.C) which used the combination of the two models(III.A and III.B) gave the best results as expected. We achieved the accuracy of **97.88%**. Therefore the ensemble method(III.C) provided us with the best results.

ACKNOWLEDGEMENT

In this technology driven world, where every day new technologies are devised, that help make the society safer and a better place, we are obliged to get the opportunity to showcase our research under the topic of detection and classification of Malwares through the medium of this paper. Apart from our hard work, we must not forget and acknowledge the efforts put in by our mentors and advisers. We would primarily like to thank Prof. Naresh K (naresh.k@vit.ac.in), our guide for this project for his immense patience and precise guidance that helped us throughout the duration of our research. Apart from his enlightening knowledge about the subject that cleared many of our doubts, he broke the formal barrier and advised us like a friend. We would also like to thank our school SCOPE, where we were allowed to use equipment and software that were crucial in the completion of our research.

REFERENCES

[1] J. Fu, J. Xue, Y. Wang, Z. Liu and C. Shan, "Malware Visualization for Fine-Grained Classification," in IEEE Access, vol. 6

[2] L. Liu and B. Wang, "Malware classification using gray-scale images and ensemble learning," 2016 3rd International Conference on Systems and Informatics (ICSAI), Shanghai, 2016

[3] F. M. Darus, N. A. A. Salleh and A. F. Mohd Ariffin, "Android Malware Detection Using Machine Learning on Image Patterns," 2018 Cyber Resilience Conference (CRC), Putrajaya, Malaysia, 2018

[4] R. Kaur, Y. Ning, H. Gonzalez and N. Stakhanova, "Unmasking Android Obfuscation Tools Using Spatial Analysis," 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, 2018

[5] Jinpei Yan, Yong Qi, Qifan Rao, "Detecting Malware with an Ensemble Method Based on Deep Neural Network", Hindawi Security and Communication Networks, 2018

[6] Ke He, Dong Seong Kim, "Malware Detection with Malware Images using Deep Learning Techniques", 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2019

[7] M. Zakeri, F. Faraji Daneshgar, M. Abbaspour, "A Static Heuristic Approach to Detecting Malware Targets", Security and Communications Networks, vol.8

[8] P. Li, L. Liu, D. Gao, M. K. Reiter, "Challenges in Evaluating Malware Clustering", International Symposium on Recent Advances in Intrusion Detection, 2010

[9] L. Nataraj, S. Karthikeyan, G. Jacob, B. S. Manjunath, "Malware Images: Visualization and Automatic Classification", International Symposium on Visualization for Cyber Security, July 2011

[10] David Kornish, Justin Geary, Victor Sansing, Soundararajan Ezekiel, "Malware Classification using Deep Convolutional Neural Networks, IEEE 3, 2018

[11] Z. H. Zhou, "Ensemble methods foundation and algorithms. Machine Learning and Pattern Recognition", Taylor & Francis, UK, 2012