International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-4, April 2020

An Effective Method for Predicting Malware Family

Nourin N.S, Sulphikar A

Abstract: ***Today, many of devices are connected to internet through networks. Malware (such as computer viruses, trojans, ransomware, and bots) has becoming a critical concern and evolving security threats to the internet users nowadays. To make legitimate users safe from these attacks, many anti-malware software products has been developed. Which provide the major defensive methods against those malwares. Due to rapid spread and easiness of generating malicious code, the number of new malware samples has dramatically increased. There need to take an immediate action against these increase in malware samples which would result in an intelligent method for malware detection. Machine learning approaches are one of the efficient choices to deal with the problem which helps to distinguish malware from benign ones. In this paper we are considering xception model for malware detection. This experiment results shows the efficiency of our proposed method, which gives 98% accuracy with malimg dataset. This paper helps network security area for their efficient works.***

Keyword: ***Convolution Neural Network, Machine Learning, Malware Detection.***

I. INTRODUCTION

In the internet age, there are more possibility to take place malicious actions (such as encrypting data, hijacking etc.). For the computer users, the security of computer system become more concern. To avoid those attacks many anti-malware defensive methods came up. Which includes scanners, signature-based techniques, software. The scanners are the traditional ones which has many demerits that will be covered in signature-based methods. As the years goes new researches increases in the machine learning field. The rapidity in increase of malware samples or malicious attacks cause the exponential growth in defensive methods. Users needs their security from those attacks which are taking place due those malicious actions. Malware detection methods are mainly separated into two which is clearly depicted in Figure 1. First is signature-based method in which signatures are crosschecked between the suspicious file and signatures in the database. The main signature methods are hash signature method and byte signature method. Next method is Heuristic-based method which are mainly divided into static technique and dynamic technique.

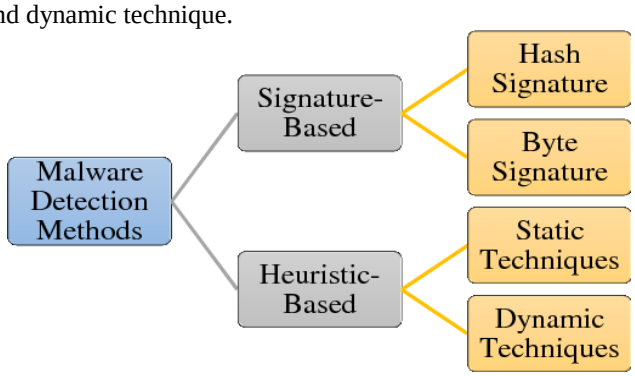

Fig.1. Different malware detection methods [1].

Static analysis and dynamic analysis methods covers all the limitations in the signature methods. Static technique is fast and safest method which are good at analyzing multipath malwares. It has a low level of false positive that shows the analysis having a high accuracy rate and efficient. The limitation faced by static method is, it cannot analyze obfuscate malware. Dynamic analysis or behavior analysis will execute suspicious files in supervised environment. When compared to static method it is time consuming due to the execution. Later on, new approaches came up with the idea of integrated static analysis and dynamic analysis. Hence generated an efficient method by combining the advantages of both methods [8,9]. Gradually machine learning methods become significant and those methods are applied for the classification techniques. There comes the Naive Bayesian, Decision Tree, Random Forest, Support Vector Machine methods. by using these methods an accurate malware classification can be achieved.

II. RELATED WORKS

Many researches are done on malware detection area. Detection Methods can be distinguished in many ways which are truly based on the point of view. Signature-based method extract unique signature from malware files and with those signature suspicious files are cross crosschecked to detect whether it is malware or not. In Domodaran et al. [2] experiments a signature-based method is proposed in which byte sequence or a hash file is used as signature to detect the malware. In Souri et al. [3]

Revised Manuscript Received on April 15, 2020.
* Correspondence Author
Nourin N.S*, Department, Name of the affiliated College or University/Industry, City, Country. Email: xyz1@blueeyesintlligence.org
Sulphikar A, department, Name of the affiliated College or University/Industry, City, Country. Email: xyz2@blueeyesintlligence.org

illustrated a signature method for identification of malware. It extracts byte sequence as marks by implementing static analysis technique. In the case of static analysis, it usually analyzes Portable Executable files. Without executing them. The patterns which are used for this analysis can be extracted in many forms. In Gandota et al. [4] determine the patterns like API calls, signatures in strings, frequency in operation code etc. Ucci et al. [5] uses control flow graph as patterns in which block of code are represented as nodes and flow path as edges thereby capture file behavior and acquires program structure. Dynamic analysis approach mainly focused on the API calls to represent malware behavior. Ki et al. [6] extracts user level API calls. Latterly, for similarity matching sequence they used Longest Common Subsequence algorithm which acquires 98% accuracy. Decision Tree used by Galal et al. [7] also uses API calls for capturing information which are relevant for malware analysis. This method also achieved a detection rate of 97.19%.

III. BACKGROUND

Convolution neural network (CNN) is a special type of neural network which has proven its efficiency in different areas such as classification, image recognition etc. CNN has different types of topologies which includes Residual Network (ResNet), VGG-16, Xception etc. In this paper we are introducing Xception, a special CNN topology for malware classification problem.

CNN architecture consists of several layers which includes:

i. **Input Layer** where inputting the raw pixel values such as the width, height and the three colour channels (R, G, B) of the image.
ii. **Convolution Layer (CONV)** here certain features are extracted from the input image which is placed into a set of convolution filters.
iii. **RELU Layer** also known as activation layer. In which activated features are transmitted to the next layer.
iv. **Pooling Layer** mainly deals with down-sampling process. Which helps to reduce memory usage.
v. **Fully-Connected Layer** helps in the flattening process. That means it flatten the results acquired from the previously layer.

In the case of Xception topology the input format is of 299×299 RGB image. It contains 36 convolution layers for extracting features and has a depth of 126. Instead of fully-connected layer here uses an average pooling layer to reduce the number of parameters. The convolution layers are structured into 14 modules and then divided into 3 main parts. First part is the entry flow where the data first pass through, which contains 8 convolutional layers. Then the middle section comes with the last 24 layers and finally the bottom part has 4 layers to be judged. The Xception reduce the convolution operation cost.

IV. METHODOLOGY

Our approach is divided into three phases. In the first steps a model is extracted and in the next step the model is trained with the output signals. Here we are using Xception model for our training method. It eventually ends up by producing confusion matrix of different malware families.

DATA SET USED

Dataset here used are the malimg dataset [10]. Which consist of 9,339 malware samples of 25 malware families. These malimg dataset are in the form of gray scale images, converted from malware binaries.

PREPROCESSING

The malimg dataset consist of malware samples are processed first. Figure 2 shows the malware samples of Malimg dataset which are processed.

```
Label: 0    Family:      Adialer.C Number of images: 122
Label: 1    Family:      Agent.FYI Number of images: 116
Label: 2    Family:      Allaple.A Number of images: 2949
Label: 3    Family:      Allaple.L Number of images: 1591
Label: 4    Family:  Alueron.gen!J Number of images: 198
Label: 5    Family:      Autorun.K Number of images: 106
Label: 6    Family:    C2LOP.gen!g Number of images: 200
Label: 7    Family:        C2LOP.P Number of images: 146
Label: 8    Family: Dialplatform.B Number of images: 177
Label: 9    Family:      Dontovo.A Number of images: 162
Label:10    Family:       Fakerean Number of images: 381
Label:11    Family:  Instantaccess Number of images: 431
Label:12    Family:     Lolyda.AA1 Number of images: 213
Label:13    Family:     Lolyda.AA2 Number of images: 184
Label:14    Family:     Lolyda.AA3 Number of images: 123
Label:15    Family:      Lolyda.AT Number of images: 159
Label:16    Family:    Malex.gen!J Number of images: 136
Label:17    Family:  Obfuscator.AD Number of images: 142
Label:18    Family:       Rbot!gen Number of images: 158
Label:19    Family:     Skintrim.N Number of images: 80
Label:20    Family:  Swizzor.gen!E Number of images: 128
Label:21    Family:  Swizzor.gen!I Number of images: 132
Label:22    Family:          VB.AT Number of images: 408
Label:23    Family:     Wintrim.BX Number of images: 97
Label:24    Family:        Yuner.A Number of images: 800
Processing images ...
Images processed: 9339
```

Fig 2. Malimg dataset processed-family and their corresponding image counts.

ARCHITECTURE

The architecture in Figure 3 shows the overview of proposed system. In which the input malimg dataset is processed and the features are extracted. With the help of features modeled is trained and ready for the testing. Finally, after k-fold validation and testing we get an accuracy which is efficient for the classification and also generating confusion matrix.

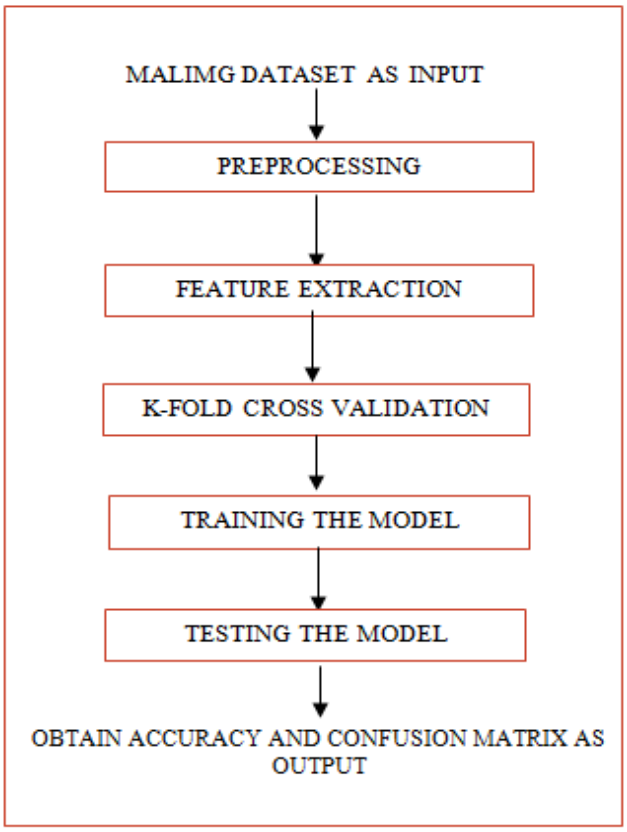

Fig.3. Architecture of proposed system

COMPUTATIONAL MODEL

Here using Xception model for the training purpose. As we know that the input format for images in Xception is of 299 × 299. In this experiment, setting the input size of our base model as 224 × 244. Then applying maxpooling for the extraction of xception features. These features help the model to train accurately and predict the malware family effectively.

K-fold cross validation is applied for training and testing test. Each iteration is usually named as folds, here setting number of folds to 10 (k=10). In the Figure 4, clearly depicts the 10-fold cross validation by dividing the training set into 10 groups. From those 10 groups use one of them to train the model. The exponential increase helps in overall improvement of outcome for the process. After the 10 iterations we get test accuracy for corresponding 10 iterations. From that calculate average accuracy of the model.

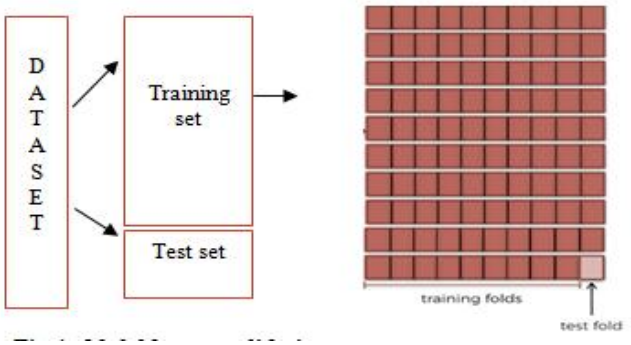

Fig.4. 10-fold cross validation

V. RESULT AND DISCUSSIONS

Dataset has partitioned for testing and training. Then preprocess malimg dataset that contains 9339 malware samples. After the 10 iterations of testing, we get an average testing accuracy of 98%. Which proves that it will be a one of the best methods for the classification. Finally, plotting a confusion matrix for the malware family which gives the picture of how close the predicted and actual value relates Figure 5 shows the confusion matrix of the test results, which shows the prediction accuracy of each malware family in Figure 2.

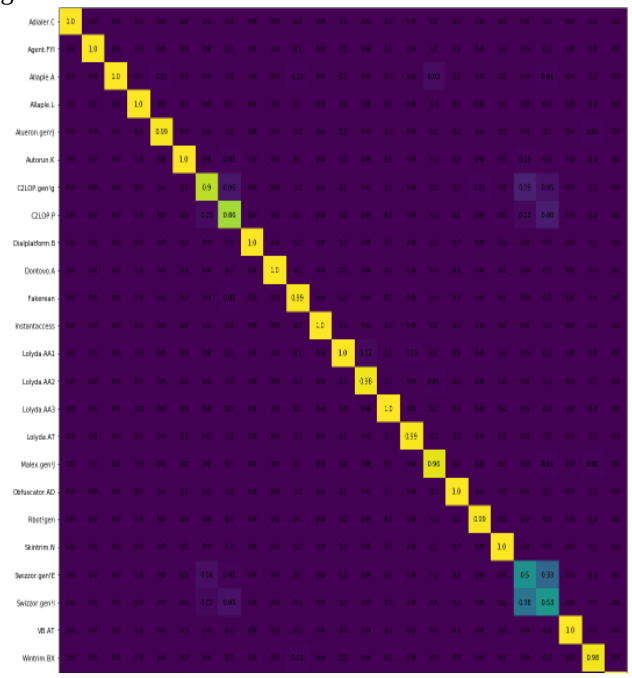

Fig.5. Confusion matrix of malimg dataset

VI. CONCLUSION

The researchers extend their approaches day by day in the field of malware detection. Each year there is a dramatic increase in millions of malwares. To defense against those malware attacks the users have to take corresponding counter measures. For that new detection methods with high detection rate are developed. That gives users a malware free environment up to an extends. In this paper we have employed a method to predict the malware family in an efficient way. Classified each malware into its corresponding family and generated confusion matrix.

REFERENCES

1. Sihwail, R.; Omar, K.; Ariffin, K.A.Z. A Survey on Malware Analysis Techniques: Static. Dyn. Hybrid Mem. Anal. 2018, 8, 1662–1671.
2. A. Damodaran, F. Di Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," J. Comput. Virol. Hacking Tech., vol. 13, no. 1, pp. 1–12, 2017.
3. A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," Human-centric Computing and Information Sciences, vol. 8, no. 1. 2018.
4. E. Gandotra, D. Bansal, and S. Sofat, "Malware Analysis and Classification: A Survey," J. Inf. Secur., vol. 05, no. 02, pp. 56–64, 2014.

5. D. Ucci, L. Aniello, and R. Baldoni, “Survey on the Usage of Machine Learning Techniques for Malware Analysis,” arXiv Prepr. arXiv1710.08189, pp. 1–67, 2018.
6. Y. Ki, E. Kim, and H. K. Kim, “A novel approach to detect malware based on API call sequence analysis,” Int. J. Distrib. Sens. Networks, vol. 2015, no. 6: 659101, pp. 1–9, 2015.
7. H. S. Galal, Y. B. Mahdy, and M. A. Atiea, “Behavior-based features model for malware detection,” J. Comput. Virol. Hacking Tech., vol. 12, no. 2, pp. 59–67, 2016.
8. M. Eskandari, Z. Khorshidpour, and S. Hashemi, “HDM-Analyser: a hybrid analysis approach based on data mining techniques for malware detection,” J. Comput. Virol. Hacking Tech., vol. 9, no. 2, pp. 77–93, 2013.
9. P. V. Shijo and A. Salim, “Integrated static and dynamic analysis for malware detection,” in Procedia Computer Science, 2015, vol. 46, pp. 804–811.
10. https://www.dropbox.com/s/ep8qjakfwh1rzk4/malimg_dataset.zip?dl=0

AUTHORS PROFILE

Nourin N.S is pursuing (4th Semester) Master’s Degree in Computer Science and Engineering from LBS Institute of Technology for Women, Kerala, India affiliated under Kerala Technical University.

Sulphikar A currently, he is working as an Associate Professor in Computer Science and Engineering, LBS Institute of Technology for Women, Trivandrum, India, affiliated under Kerala Technical University.