Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 4, April 2024

Malware Detection and Classification in Portable Executable Files Using Deep Learning Methods

Jhansi Priya S, Abdul Sadiq,
Akanksh Pn, Dhruva S Kashyap
Students, Department of Information Science and Engineering
BMS Institute of Technology and Management
Bengaluru, India

Dr. M V Sudhamani
Professor, Department of Information Science and Engineering
BMS Institute of Technology and Management
Bengaluru, India

Abstract— **The aim of this work is to present an approach to detect and classify the Portable Executable (PE) files on windows operating system as Malware or Benign. In a time where network-connected devices are vulnerable to malware, early detection is critical to minimize monetary and societal harm. Leveraging the benefits of Long Short-Term Memory (LSTM) networks, and YARA rules, this work presents a comprehensive approach to extract the features from the files and uses Convolutional Neural Networks (CNN) to classify the files as either Benign or Malicious. Some samples of malware along with another dataset of 10,000 images, 7,000 images were used for training and 3,000 images were used for testing with the proposed system. For safe implementation, the work is executed in the virtual environment provided by Sandboxie. The system exhibits robustness in differentiating between benign and malicious files, with total accuracy of 94.78%, precision of 88.64%, recall of 94.76% and F1 score of 91.59%. By combining CNN and LSTM networks with YARA rule-based detection, the system demonstrates competence in detecting and categorizing malware in its early phases. Additionally, a user-friendly interface is developed to facilitate the user to submit the file to be checked for malware.**

Keywords— **CNN; RNN; LSTM; Malimg; Windows Malware; PE file; YARA rule set; Sandboxie.**

I. INTRODUCTION

Malware is any code that tampers the normal functioning of a program. Malware attacks have been on the rise, coinciding with the growing dependence of individuals on the internet and data. In 2020 a 78% rise has been observed in ransomware attacks as reported by "The State of Ransomware 2022" [2]. In 2021 FBI's Internet Crime Complaint Centre received 3,729 complaints regarding ransomware attacks. As indicated by the Business resources, the damages of worldwide cybercrime will cost over $10.5 trillion annually by 2025 [1]. Earlier malware was limited and were of specific type with standard signatures, even if new ones were created, they would resemble their parent malware so, they were easy to detect. But the next generation malwares are obfuscated, they don't use the same signatures, hence the traditional method of signature comparison won't work here and needs a more sophisticated approach. Attackers set big organizations like ecommerce websites, banking sector, healthcare industries or educational sectors as their targets as they can get access to a huge amount of valuable data. They trick the organization to click on a faulty link or nowadays opening a mail or a message is sufficient to trigger the process. Even if the attackers get access to any one field of the customer's data, it opens many doors for the attacker to gather information. For instance, let's consider the attacker got access to the registered email of the customers of an ecommerce website. Using this, the attacker can find out the websites the customer has used the same email to register. Among them all, if one website's database protection is weak and the password is revealed, there is a high possibility that the user would have used the same password in some other website. This is just a scenario to help us understand the complexity of malware attacks.

The most crucial step in protecting ourselves and the organizations is figuring out whether or not a malware is present. The process of detecting and classifying a file or a program as malware or benign based on different methods like traditional signature based and behavior based or machine learning techniques is referred to as malware analysis. The traditional methods like heuristic based approaches are the most effective ones. But as the malware is evolving these approaches have become less effective yet useful in detecting them at the first stage. In the present world, where rapid development is seen in several domains like Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and so on, there has been an increase in the development of evasive malwares, which can trick detection systems and are highly sophisticated and self-evolving.

In the recent years, numerous techniques utilizing various DL or ML algorithms have been put forth with the aim of detection. Several strategies combine ML and DL algorithms for the identification of malware which are anticipated to surface in the near future, have been created to combat the new types of malwares. It has been found that these techniques are effective in detection with a noticeably high efficiency and accuracy rate. Each approach considers different variables in order to locate, classify, and investigate malware. Each technique has its own benefits and drawbacks, and one may be more effective than the other based on the specifics. Currently there are very less methods that can identify every new

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 4, April 2024

generation of sophisticated malware, despite different kinds of strategies that have been proposed.

Some malwares can be identified by the unusual behavior of the system like excessive heating, fast battery drain, disc overload. The malware can be obfuscated and can escape the detection sometimes. To solve this problem, our research uses Yara rules, Recurrent Neural Network (RNN)-LSTM and CNN.

Performance metrics such as recall, accuracy, precision, and F1 score were used to evaluate the effectiveness of this model in classifying the PE files as malware or benign.

II. LITERATURE SURVEY

Malware analysis uses both static and dynamic approaches to assess the damage and determine the level of complexity of the intrusion. Dynamic analysis monitors the sample in a secure environment, static analysis looks at malware without running it. Present-day detection techniques, which solely depend on the traditional methods, fail to recognize unfamiliar malware in real-time. Machine learning (ML) techniques are used for more efficient malware analysis to counter the evolving evasion techniques like polymorphism. In malware detection, deep learning (DL) eliminates the need for feature engineering and improves performance.

In the experiments [3], the malware sample was run and programs like PEiD (Portable Executable iD) detector, PEview, and Wireshark is used for analysis. Designed as a Trojan, malware QQQ.exe was generated and is intended to infect Intel 386 and later processors. The file size is 140kb, and it imports ADVAPI32.dll, KERNEL32.dll, and SHELL32.dll. Malware QQQ.exe communicates distant servers, disables security systems, and uses too much RAM (Random Access Memory) once it has infected a system. It functions as ransomware, executing files dropped to %Public% and demanding a ransom on a designated bitcoin address.

The paper [4] provides a thorough examination of behavioral frequency analysis as a potent technique for distinguishing between malicious and benign applications, with a focus on Windows API system function calls. Based on API call frequencies, the study assesses the efficacy of different ML models in identifying software as malicious or benign. It employs the use of datasets, applying several algorithms such as Random Forest (RF), K-Nearest Neighbors (KNN) and Logistic Regression (LR) by tracking how well each does in classification. The outcomes demonstrate that different datasets and models used, have differing bias and accuracy, with Decision Trees giving more accurate results in some circumstances. This also emphasizes how typical Windows API calls found during dynamic malware research can be used to fingerprint malicious activities. Deeper understanding of malware behaviors and patterns are made possible by the research, which emphasizes the significance of API call frequency in differentiating between dangerous and benign software.

The work [5] presents a ML-based dynamic method to analyze the malware, with a specific emphasis on Windows-based malware. Detailed instructions are followed in the process, which starts with gathering datasets and ends with selecting and extracting features before using machine learning classifiers. The performance of the classifiers—Support Vector Machine (SVM), Naive Bayes and RF. Windows-based malware detection is demonstrated through a summary of key findings. Using 41 selected features, the Random Forest classifier and Genetic Algorithm feature selection together yielded the greatest accuracy of 86.8% among the tested classifiers.

With an aim to improve classification effectiveness and accuracy [6] suggested an approach using DL and ML models, for feature extraction methods including looking at sections, byte codes, operational codes and system calls. The key objective was to assess how well DL models performed against more conventional methods. It addressed the usage of diverse feature sets and classifiers, highlighting Deep Neural Network's (DNN), advantages in obtaining results across a range of feature sets. It also discusses malware analysis going over earlier approaches such as image processing, n-grams, static analysis and semi-supervised learning. It draws attention to the expanding use of DL methods in particular, neural networks for the detection and interpretation of unsafe executables. The proposed next step entails applying RNNs and CNNs for malware classification and evaluating the combined impact of all feature sets on loss and accuracy.

Using runtime behavioral characteristics from PE files, [7] suggested CNN-based Windows malware detector which achieves an astounding 97.968% detection accuracy. The study visualizes malware features, try out grayscale image representations, and use deep learning to effectively extract features from unprocessed data. The CNN-based Windows malware detection model is optimized by feature selection techniques like Chi Square, Information Gain, Mutual Information and Relief, which also recommend important API calls and categories for examination. The Behavior-based Feature Extractor, Feature Selector, Image Generator and CNN modules among others, are part of the suggested approach and help with malware detection and classification. The core of the approach is to extract behavioral reports from PE files, arrange them according to the Malware Instruction Set (MIST), and emphasize the significance of dynamic features such as CAT API calls. The effectiveness of the CNN-based model is demonstrated through experiments, which show high recall, accuracy, precision and F-measure, particularly when the Relief Feature Selection Technique is used as a guide. The CNN-based method is more accurate than other approaches, but it takes a little longer to detect changes than other classifiers.

The study [8] discusses several models, approaches and strategies that make use of sizable datasets such as the EMBER dataset. For the purpose of classification of malware in Windows environments, features extraction, association mining, and API call sequencing are emphasized. It emphasizes the intricacy of contemporary malware, including fileless malware and obfuscation techniques. In an environment where malware may exist, it emphasizes the importance of neural networks and DL models in handling a variety of file formats. The article covers DL based techniques like regularization strategies, activation functions, and several loss functions that reduce overfitting and boost accuracy during training.

The work [9], emphasizes how well DL handles raw binary bytes and extracts features from images. A 97% accurate model with fusion feature sets is demonstrated, and multi-view feature

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 4, April 2024

integration is suggested for dependable detection. Comprehensive analyses of diverse ML and DL frameworks for malware classification are offered. The suggested method combines many data viewpoints in a multi-view feature fusion strategy for reliable virus detection. Additionally, it thoroughly assesses ML methods and DL model architectures for malware classification. Important discoveries highlight the value of behavioral characteristics, the improved performance of some feature sets, such as PE Import, and image representation using CNN architectures.

The work [10] examines several malware detection topics, with particular emphasis on DL and ML models, methods for converting executable files into images, handling unbalanced datasets, and combining ML and DL for enhanced detection. It suggests a system that combines SVM with improved DL models, transfer learning for malware detection. Experiments with multiple datasets, including VirusShare, Malign, and Microsoft malware datasets, show how effective models like VGG16, ResNet50, InceptionV3, and MobileNet are at detecting malware. A comparative study demonstrates how accurate and efficient the suggested framework is compared to existing approaches. It discusses model performance, the effect of data augmentation, and the importance of DL models in malware detection accuracy, such as VGG16, VGG19, and ResNet50.

In summary, the literature survey highlights the progress made in the detecting model from heuristic based approach to DL models. DL techniques are used to identify the malware through image, by converting the PE files to grayscale image. The incorporation of different ML strategies and sophisticated DL architectures, along with the traditional methods is a faster approach. ML models including CNNs, SVMs, RFs and ensemble models, showcases higher accuracy rates, but with a drawback of overfitting. Despite persistent challenges such as limited updated datasets and more generalizability required, the ongoing evolution of these new practices holds great promise in transforming. The survey notably highlights diverse approaches encompassing feature extraction, combination of DNNs and CNNs, ML optimization and the deployment of advanced neural networks to analyze the PE file.

III. PROPOSED WORK

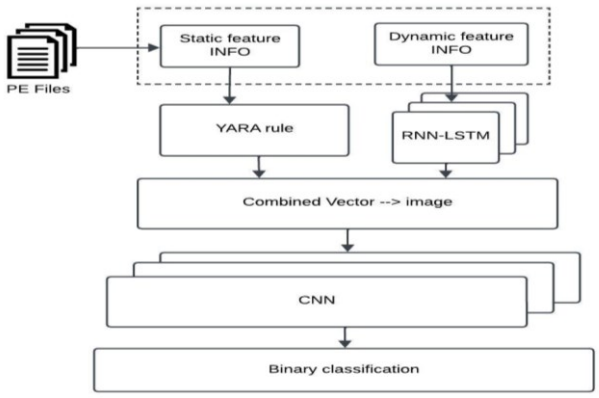

Fig. 1. Proposed System Architecture

The system architecture for the proposed work consists of five modules namely Data Acquisition, Preprocessing, Feature Extraction, Feature Encoding and Classification as shown in *Fig. 1*. These methods are carried out in the sandbox environment provided by Sandboxie. The sandbox environment is a virtual environment used to run malware by protecting our device and files.

A. Data Acquisition

This project work considers an integration of different datasets. **"**Windows Malware**"** is one of the popular datasets, consisting of malware samples for windows operating systems. It contains datasets for different features of malware. The features considered in this work are PE Header, Assembly Instructions and Imports Functions. Some samples of this dataset are used in training this model. Another most popular dataset for images of malware is Malimg, consisting of grayscale images of only malware PE files. It has 25 families of malware images. The benign dataset considered is custom consisting of 3000 benign images.

B. Data Preprocessing

Since only PE files of malware are not available, this stage filters out the files that are of formats other than .exe. It also calculates the hash of the malware file and identifies the number of sections in the PE file and passes it to the next stage.

C. Feature Extraction

YARA rule set plays a major role in analyzing the code statically. YARA rules are a predefined set of rules that look for certain patterns in the PE files. The patterns can be text strings, API calls for debuggers, signs of any packers used to compress the code. The files that showed positive signs are declared as malware. The files that showed negative results are moved to the next phase, i.e., the dynamic feature learning.

A specific type of RNN, i.e., the LSTM model is used to train the feature extraction model with a labelled dataset. Each feature extracted are stored in a respective file belongs to different data items in a dataset. The features extracted are preprocessed and is fed in sequence to the LSTM model to learn the dynamic features. The LSTM looks at the current input and the information it gathered from earlier inputs to decide what to remember from the features.

D. Feature Encoding

The feature vector obtained from the LSTM is converted into a linear binary vector using OneHotEncoder. This encoding helps to convert the feature map into a grayscale image. The encoded vector is converted into a grayscale image which is fed into CNN model for further process.

E. Classification

CNN images have found to give promising results in image processing.

- Input Layer: Receives the preprocessed image.
- Two Hidden Layers:
- Convolutional Layers (conv2d): Consisting of 16 filters of size 5x5 filter in the first layer and 3x3 filter in the second layer.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 4, April 2024

- Max pooling Layers: It reduces the size of the image by taking the maximum of the value, basically abstracting it.
- Dropout Layers: This helps preventing overfitting.
- Flatten Layer: This layer transforms multidimensional feature maps into a 1D vector.
- Fully Connected Layer: Perform classification of the output from the vector received from the flatten layer.
- Output Layer: Provides the classification result (the probability of being malware).

IV. RESULTS

The feature extraction from the PE files and conversion of the feature vector into a grayscale image followed by the classification of files as malicious or benign, yields critical insights of how obfuscated the malware can be.

In the evaluation of the training process, a confusion matrix serves as a fundamental tool for assessing the performance of the classification model. In this specific scenario, where 7000 training samples were utilized, comprising 4,900 malicious and 2,100 benign files, the confusion matrix provides a concise summary of the model's predictions. It delineates four essential metrics: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). True positives represent correctly identified malicious files, while true negatives denote accurately classified benign files. False positives signify benign files cases erroneously classified as malicious files, and false negatives indicate malicious files misclassified as benign files. With an accuracy of 94.78%, precision of 88.64%, and recall of 94.76%, the model demonstrates strong performance in accurately classifying instances of malicious files and benign files cases during training this is given in *Fig. 2*.

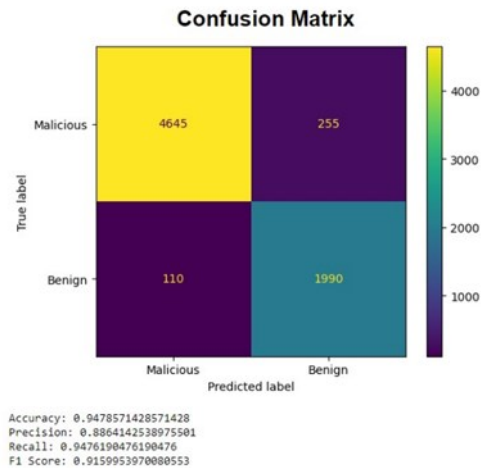

Fig. 2. Confusion Matrix of Pretrained Model

On completion of feature extraction and training of the model, a user interface is designed to enable the uploading of files for classification.

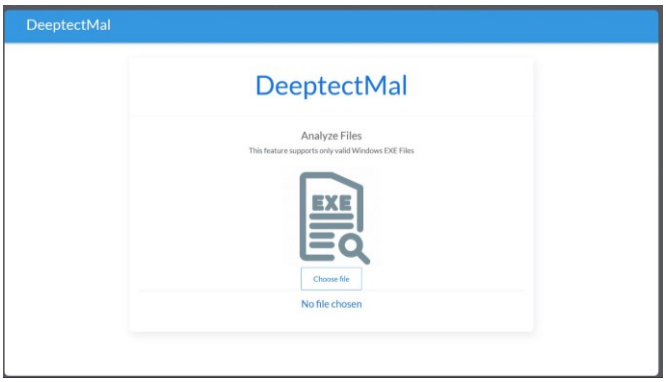

Fig. 3. Interface to upload a file from the system

Fig. 3, gives the preview page designed for users to upload a file for classification within the malware detection and classification system. This page offers a user-friendly interface, allowing individuals to easily select and upload the suspicious PE file.

Some of the intermediate results obtained while processing the files for classification are shown below. *Fig. 4*, shows the uploading of a benign file.

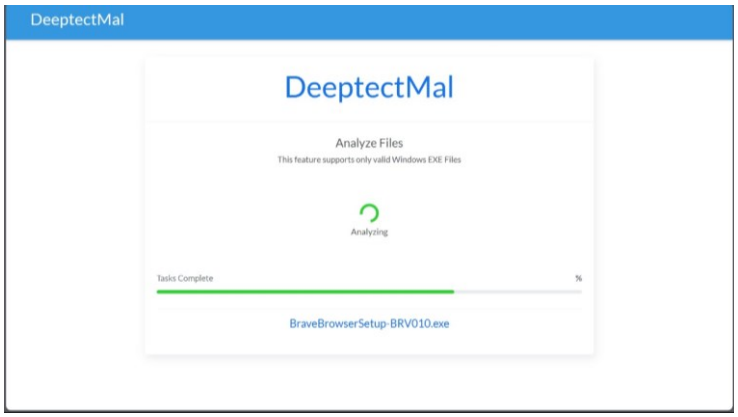

Fig. 4. Upload of Benign File

Fig. 5, shows the prediction page, where the uploaded PE file undergoes classification within the malware detection and classification system.

Fig. 5. Classification as Benign

Fig. 6, shows the uploading and processing of a malicious PE file downloaded from an untrusted website.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 4, April 2024

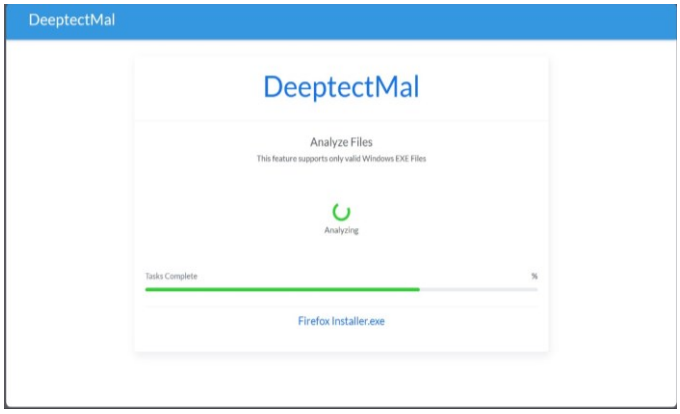

Fig. 6. Upload of a malicious file

Fig. 7, shows the results of the file classified as malicious.

Fig. 7. Classification as Malicious

Along with the classification results, the percentage of Maliciousness of the file is also displayed. *Fig. 8*, shows the percentage of maliciousness in the file.

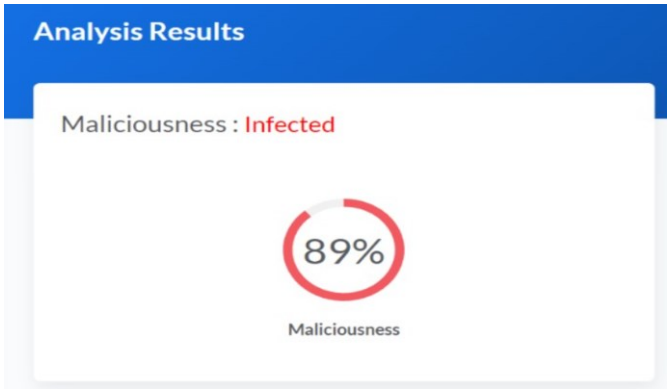

Fig. 8. Percentage of Maliciousness in a Malicious file

V. CONCLUSION

The proposed work represents a significant advancement in the field of malware detection and classification. By leveraging DL techniques and using a combination of different datasets and comprising 10,000 images, including both training and testing subsets, this work demonstrates a comprehensive approach to malware detection and classification. Through the integration of sophisticated feature extraction, feature encoding methods, LSTM and CNN, the system effectively recognizes the patterns found in malicious files. The development of a user-friendly interface further enhances the usability and efficiency of the system. With an accuracy of 94.78%, precision of 88.64%, and recall of 94.76%, the model demonstrates good performance in accurately classifying files as malicious or benign.

In future with the usage of advanced DL model like VGGNets, ResNets, MobileNet and InceptionNets the performance can be further improved.

REFERENCES

[1] L. Seitz, 2024 Cybersecurity Almanac: 100 Facts, Figures, Predictions, And Statistics. Cybersecurity Ventures, https://www.broadband search.net/blog/alarming-cybercrime-statistics.

[2] S. M. Kerner, Ransomware trends, statistics, and facts in 2023, TechTarget, https://www. techtarget.com/searchsecurity/feature/ Ransom ware-trends-statistics-and-facts, 2023.

[3] Saurabh, "Advance Malware Analysis Using Static and Dynamic Methodology," 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India, pp. 1-5, doi: 10.1109/ICACAT.201 8.8933769, 2018

[4] A. Walker and S. Sengupta, "Insights into Malware Detection via Behavioral Frequency Analysis Using Machine Learning," MILCOM 2019 - 2019 IEEE Military Communications, Conference (MILCOM), Norfolk, VA, USA, pp. 1-6, doi:10.1109/MILCOM47813.2019.9021034M , 2019.

[5] Irshad, R. Maurya, M. K. Dutta, R. Burget and V. Uher, "Feature Optimization for Run Time Analysis of Malware in Windows Operating System using Machine Learning Approach," 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, pp. 255-260, doi: 10.1109/TSP.2019.87 68808, 2019.

[6] R. Patil and W. Deng, "Malware Analysis using Machine Learning and Deep Learning techniques," 2020 SoutheastCon, Raleigh, NC, USA, pp. 1-7, doi: 0.1109/SoutheastCon 44009.2020.936 8268, 2020.

[7] S. D. S.L and J. C.D, "Windows Malware Detector Using Convolutional Neural Network Based on Visualization Images," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 1057-1069, doi: 10.1109/TETC.2019.2910086,2021.

[8] Usha Divakarla, K Hemant Kumar Reddy, K Chandrasekaran, A Novel Approach towards Windows Malware Detection System Using Deep Neural Networks, Procedia Computer Science, Volume 215, 2022, Pages 148-157, ISSN 1877-0509, https://doi.org/10.1016/j.proc s.2022.12.017, 2022.

[9] Rajasekhar Chaganti, Vinayakumar Ravi, Tuan D. Pham, A multi-view feature fusion approach for effective malware classification using Deep Learning, Journal of Information Security and Applications, Volume 72, 103402, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2022.1034 0 2, 2023.

[10] Kamran Shaukat, Suhuai Luo, Vijay Varadharajan, A novel deep learning-based approach for malware detection, Engineering Applications of Artificial Intelligence, Volume 122, 106030, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2023.1 06030,2023.