Enhancing Malware Detection Efficiency through CNN-Based Image Classification in a User-Friendly Web Portal

Aum Shiva Rama Bishoyi[1], Vijayakumar P[2]*

[1] Undergraduate Student, School of Electronics Engineering ,Vellore Institute of Technology, Chennai, Tamilnadu, India; [2] Professor, School of Electronics Engineering, Vellore Institute of Technology, Chennai, Tamilnadu, India
vijayrgcet@gmail.com

Abstract: Traditional anti-virus programs rely on signature-based methods, which are time-consuming and ineffective for detecting encrypted or previously unseen malware. To address this, we propose a CNN-based malware detection web portal that classifies malware images using a unique dataset. Unlike signature-based methods, our approach does not rely solely on historical signatures, making it more effective for detecting new and encrypted malware. By simulating malware behavior and matching it to new programs, our CNN-based approach offers a dynamic and efficient solution for malware detection. The user-friendly interface of our web portal facilitates easy deployment of the deep-learning-based algorithm, enabling users to upload files for testing and classification with ease.

Keywords: Malware Dataset; Classification; Malware to Image Conversion; Malware Detection Web Portal; Convolutional Neural Network.

Introduction

The current landscape of malware detection is plagued by limitations inherent in traditional methodologies, primarily signature-based and heuristic-based approaches. While these methods have proven effective in identifying known malware variants, they falter when confronted with new, previously undiscovered threats. Signature-based techniques, which compare files against a predefined list of known malicious signatures, and heuristic-based approaches, which analyze program behavior rather than relying on signatures, both exhibit efficiency in detecting well-known malware strains[11]. However, they struggle to keep pace with the ever-evolving tactics of cybercriminals, particularly when it comes to identifying novel or encrypted malware. This gap in existing malware detection strategies underscores the critical need for innovative research and methodologies in the field. Despite the significant efforts invested in developing detection techniques, no approach has emerged as a panacea capable of identifying all malware in the wild. The escalation of cyberattacks, exemplified by recent instances of ransomware and social engineering tactics targeting both individuals and organizations, underscores the urgency of fortifying systems against malicious intrusions.

In this context, the proposed CNN-based malware detection system represents a significant advancement in the field. By leveraging convolutional neural networks trained on a unique dataset of

malware images, this system transcends the limitations of traditional methodologies. The conversion of malware files into images enables the CNN model to discern patterns and similarities among malware families, thereby facilitating the detection of both new and previously unseen threats. The integration of this innovative approach into a user-friendly web portal enhances accessibility and usability, empowering regular users to analyze and detect malware with ease. The recent surge in cybercriminal activity during the COVID-19 pandemic serves as a poignant reminder of the pressing need for robust malware detection capabilities. Instances such as the CovidLock ransomware and social engineering attacks underscore the devastating consequences of malware infections, ranging from data encryption and extortion to network-wide compromises. Traditional signature-based techniques have proven inadequate in detecting such sophisticated threats, emphasizing the imperative of embracing advanced detection methodologies like the proposed CNN-based approach.

In essence, the evolution of malware threats necessitates a corresponding evolution in detection strategies. By bridging the gap between existing methodologies and emerging threats, innovative approaches like CNN-based malware detection offer a promising avenue for safeguarding systems and mitigating the pervasive risks posed by malicious actors

Related work

Moser et al. [1] examined all limitations of the techniques for static analysis. They revealed that static analysis by itself is insufficient to identify or categorize malware by introducing a code obfuscation strategy. Additionally, they suggested that because dynamic analysis is less susceptible to code obfuscation conversion, it should be used in conjunction with static analysis. Static analysis has been used in numerous studies to identify malware using precise de-compilation, likeness evaluation framework, register matter using multifaceted binary programme characteristics, subroutine-based recognition, statistical analysis of assembly commands, record resemblance diagrams, de-anonymizing software developers via code stylometry, depending on a wavelet package method, assessment and contrast of disassemblers to support opcode, and more [2].With the use of static analysis, we can carry out in-depth analyses that let us determine a program's capabilities and the kinds of residues it leaves behind on the system. It is frequently utilized in several industries, including software piracy detection and malware detection. Without running any programmes, static analysis reveals comprehensive details about the design of computer software [3]. It becomes crucial for malware analysis as a result. Many useful pieces of information, such as whether any packers were used on the suspect programme, which routines are imported, and how many components the programme contains, can be gleaned by using static analysis. Alazab et al. [4] proposed a method for classifying hazardous mobile apps based on a pair of characteristics: the permissions listed in the apps and the calls made to the API. In order to calculate the degree of commonality across the two virus families, four separate hash functions are used. In order to determine the infection likelihood, a cumulative score is generated depending on how every one of the four functions were carried out.

With typical accuracy levels of 99.264% and 97.364%, respectively, it converts binaries into Markov images upon both of the prominent malware datasets, the Microsoft dataset and the Drebin dataset. Yuan et al. [5] proposed a classification technique based on byte-level deep learning and Markov pictures. To discriminate between harmful and innocuous data, Yoo et al. [6] stated an amalgamated solution

integrating a Random forest classifier with deep learning models comprising 12 hidden layers. The rate of discovery for this model was 85.1%. In other words, a behavioral model is created to recognize dangerous code. Despite producing unreliable results, these behavior models had better detection results. Onwuzurike et al. created a mechanism to assess the efficacy of static, dynamic, and hybrid analytical approaches [7]. Using datasets made up of beneficial apps from PlayDrone and the Google Play Store and bad apps from VirusShare, Lei et al. [8] obtained an F1-measure of 99% for IoT devices. Both user-generated inputs and pseudo-random input generators were used to instantiate the developed system, known as AUNTIEDROID. The model's detection accuracy is 97.2% for the BIG 2015 dataset and 98.65% for the malimg dataset, respectively. Tuncer et al.'s [9] cognitive and adaptive anti-malware model was presented to classify data from the dataset. After gathering relevant attributes using Principal Component Analysis (PCA), Linear Discriminant Analysis was utilized to develop a classification model (LDA). A method that minimizes model complexity without necessitating backpropagation or hyperparameter change was proposed by Roseline et al. [10].

Proposed CNN-Based Malware Detection using Dataset

The proposed system is built with the intention of removing above limitations. In order to overcome the limitations mentioned previously, a Convolutional Neural Network Based Malware Detection algorithm is proposed: The self-made dataset is manufactured by first downloading raw malware files in an isolated environment. The files are then converted into grayscale image with each region representing different code layers of the specific malware file. Since each family of malwares have somewhat similar layer compositions, it's hence possible to classify an unknown malware file using a model trained from such a dataset. The bits are converted to color (grayscale) intensities of values ranging from 0 to 255. The CNN model will be trained using such a dataset comprising of 4 or more classes. Therefore, it can be said that the convolutional neural network algorithm can identify features in the images which can distinguish a malware file from a benign file by converting its binary form into pixels of an image. This model works on identifying the malware file in question using a similarity-based approach rather than the conventional hashing method employed by the numerous popular anti-virus softwares. These depend on hash-tables composed of previous known malware samples; meaning, if the hash-value of the malware file exists in the hash-table, it's identified as a malignant file; else it's identified as harmless. Such a conventional methodology doesn't account for newly available malwares that might resemble their predecessors but are not exactly similar. The proposed Deep-learning (CNN) based Malware Detection model will be implemented via a web portal having a simple user interface. The portal will, in addition to carrying out the detection-classification feature on the uploaded malware '.exe' file, convey the degree of threat the file might pose to the typical host system. As shown in Figure 1, the proposed web portal will be designed to have a very simple user-friendly interface, similar to other sites that provide services on user files and have a browse file/drag-and-drop upload feature.

An image must be categorized as benign or malicious within seconds because malware detection is performed in real time. So, it makes sense to keep the image generating process quick and easy to save time. Therefore, it can be said that the proposed convolutional neural network algorithm can identify features in the images which can distinguish a malware file from a benign file by converting its binary form into pixel of an image. Instead of doing a binary classification, the data might be divided into several categories of malware or benign class as a direction for future growth.

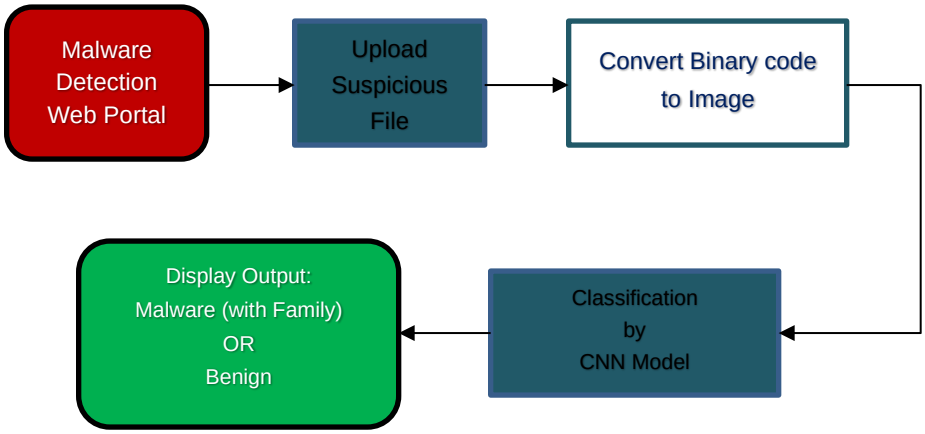

Figure 1. Block Diagram of Malware Detection Web Portal

Result & Discussion

To test this model, the initial dataset used was the ready-made MalImg dataset, which had a total of 25 different malware families. This resulted in a train accuracy of 95%, test accuracy of 95%. The process of creating the images must be speedy because detecting the malware must be finished right away. The datasets were made for the "AgentTesla," "Socelars," "Loki," and "SmokerLoader" malware families. An algorithm that employs convolutional neural networks is then applied to the images. Later, the detection model was trained using a wholly original dataset. 95% F1 score, 95% recall, 95% precision, 99% train accuracy, 100% test sensitivity at specificity, and 98% train sensitivity at specificity are reached as performance metrics. Learning curves have been plotted and shown in Figure 2 & 3. The final accuracy of our model after training and evaluation is 95%. It is converted from its binary form into an image pixel to achieve the portrayed result. Ultimately the model successfully, with an accuracy of 100%, classified an unknown malware file sample image as under the malware family 'AgentTesla'. The proposed methodology's the use of deep learning and CNNs, as opposed to more traditional methods, to recognize malware images for classification is a crucial advantage. When a model's loss function is assessed on a test dataset using the same parameters that produced the ideal function, a learning curve (also known as a training curve) is used for determining the best possible value of the function for the training dataset.

Figure 2. Learning Curve.

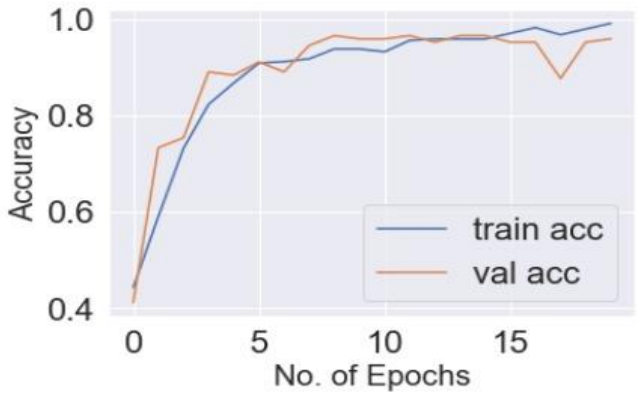

Figure 3. Accuracy over each Epoch.

Conclusion

Malware detection is crucial because it acts as a form of early alert for computer protection concerning malware and cyberattacks, which is essential given the prevalence of malware on the Internet. A deep learning model that can identify malware has been put in place to accomplish this. The malware files' binary form is changed into pictures during implementation. The dataset produced by this procedure is then used by the malware detection algorithm to categorize malware. This is a multi-classification algorithm to identify different families of malware, with the assumption that different malware file variants will have distinctive pictures.

Future research: The created framework can scale up to analyze even more malware by adding an another few layers to the current designs, which allows it to analyze a big number of viruses in real-time. The spatial pyramid pooling (SPP) layer could be utilized in upcoming work to accept input photos of any size. In this experiment, the malwares were flattened after being converted into fixed-sized images. Malware pictures can also be created using various dimensions.

References

1. Arora, S. K. Peddoju, and M. Conti, “Permpair: Android malware detection using permission pairs,” IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1968-1982, October 2019.
2. M. Nar, A. G. Kakisim, M. N. Yavuz, and ˙I. Sogukpinar, “Analysis and comparison of dis-assemblers for opcode-based malware analysis,” in 2019 4th International Conference on Computer Science and Engineering (UBMK), pp. 17–22, IEEE, November 2019.
3. O. Rawashdeh, A. Ralescu, D. Kapp, T. Kebede, "Single Property Feature Selection applied to Malware Detection", NAECON 2021 - IEEE National Aerospace and Electronics Conference, pp. 98-105, August 2021.
4. M. Alazab, A. Shalaginov, A. Mesleh, A.W. Awajan, “Intelligent mobile malware detection using permission requests and API calls”, Future Generation Computer Systems., vol. 107, June 2020.
5. Yuan B., Wang J., Liu D., Guo W., Wua P., Bao X., “Byte-level Malware Classification Based on Markov Images and Deep Learning,” Computers Security, vol. 92, May 2020.
6. Yoo S., Kim S., Kim S., Kang B., “AI-hydra: Advanced hybrid approach using random forest and deep learning for malware classification,” Information Sciences, vol. 546, pp. 420–435, February 2021.

7. Onwuzurike L., Almeida M., Mariconti E., Blackburn J., Stringhini G., de Cristofaro E., “A Family of Droids -- Android Malware Detection via Behavioral Modeling: Static vs Dynamic Analysis”, vol. 3, October 2019.
8. Lei T., Qin Z., Wang Z., Li Q., Ye D., “EveDroid: Event-aware Android malware detection against model degrading for IoT devices,” IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6668- 6680, August 2019.
9. Tuncer T., Ertam F., Dogan S., “Automated malware identification method using image descriptors and singular value decomposition,” Springer Multimedia Tools and Applications, vol. 80, pp. 10881–10900, January 2021.
10. Roseline S. A., Geetha S., Kadry S., Nam Y., “Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm,” IEEE Access, vol. 8, pp. 206303–206324, November 2020.
11. OmaMageswari,M., Vijayakumar,P.(2023).” Advancements in Technology and Techniques: Comprehensive Survey on Artificial Intelligence-Based Malware Analysis and Detection Methods”. Engineering, Science, and Sustainability, 1st Edition, CRC Press Taylor and Francis. https://doi.org/10.4324/9781003388982.