ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

ADVANCED MALICIOUS APPLICATION DETECTION USING DEEP LEARNING

M Sarojini Rani[1]
Asst. Professor
Department of CSE(DS)
Tkr College of Engineering & Technology
sarojinirani@gmail.com

T Charitha [2]
B.Tech(Scholar)
Department of CSE(DS)
Tkr College of Engineering & Technology
charitha0722@gmail.com

V Aravind [3]
B.Tech(Scholar)
Department of CSE(DS)
Tkr College of Engineering & Technology
arivindrebal630@gmail.com

S Pranav[4]
B.Tech(Scholar)
Department of CSE(DS)
Tkr College of Engineering & Technology
sangichettypranav@gmail.com

K Venu [5]
B.Tech(Scholar)
Department of CSE(DS)
Tkr College of Engineering & Technology
kvenuyadav91@gmail.com

ABSTRACT

Detecting and classifying malicious software ,or malware, is not an easy job, and there isn't a foolproof way to do it. Funding standard benchmarks for malware detection is harder than in many other research fields. This paper looks into the latest improvements in detecting malware on various platforms , including MacOS , Windows, iOS, Android, and Linux. The pre-trained and multi-task learning models for malware detection approaches toobtain high accuracy and which the best approach if we have a standard benchmark dataset. We discussthe issues and the challenges in malware detection using DL classifiers by reviewing the effectiveness of theseDL classifiers and their inability to explain their decisions and actions to DL developers presenting the needto use Explainable Machine Learning (XAI) or Interpretable Machine Learning (IML) programs. Additionally,we discuss the impact of adversarial attacks on deep learning models, negatively affecting their generalizationcapabilities and resulting in poor performance on unseen data. We think it's important to train and evaluate how well the latest deep learning models work with different malware datasets .In our study, we look at eight well-known deep learning methods across several datasets. This survey will help researchers develop ageneral understanding of malware recognition using deep learning.

I. INTRODUCTION

Operating systems such as Windows, Android, Linux, and MacOS areupdated every few weeks to protect against critical vulnerabilities. At the same time,creators of malware are constantly searching for fresh methods to adapt their harmful software to bypass the latest updates in operating system. Every operating system is vulnerable. In addition, since operating systems run on desktops and servers, and even on routers, securitycameras, drones and other devices, the biggest problem is we need a variety of systems to keep everything safe since each device is quite unique. For example, In October 2022,a hacker group from Russia called Killnet launched cyberattacks against the government services of several states, including Colorado ,Alabama, Alaska, Delaware, Connecticut, Mississippi and Kansas websites.1 Again in 2022, hackersworking on behalf of the Chinese government stole $20 million fromcovid relief benefits.2 The increase in the vulnerability of sensitive datadue to cyber-attacks, cyber-threats, cyber-crimes, and malware needs tobe countered. In 2023, Fig. 1 shows countries that

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

have been attackedby malware and the top origins of these malware.Researchers have used deep learning to classify malware samplessince it generalizes well to unseen data. Our survey focuses on static,dynamic and hybrid malware detection methods in Windows, Android,Linux, MacOS, and iOS. We talk about the good and bad sides of deep learning models for malware detection. Most recent research usesdeep neural networks (DNNs) for malware classification and achieveshigh success. State-of-the-art DNN models have been developed againstmodern malware such as Zeus, Fleeceware, RaaS, Mount Locker, REvil,LockBit, Cryptesla, Snugy, and Shlayer.

II. RELATED WORK

Deep learning has become a prominent approach in malware detection, offering robust solutions for evolving cyber threats. Static analysis techniques employ deep learning models like Convolutional Neural Networks (CNNs) to analyze malware binaries as raw data or visual images, while dynamic analysis leverages Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to monitor behavioral patterns, such as system calls and file activities. Hybrid approaches combining static and dynamic data have enhanced detection rates. However, adversarial attacks present challenges, prompting research into improving model robustness against obfuscation techniques.

Graph-based models like Graph Neural Networks (GNNs) utilize structural representations of malware, while Generative Adversarial Networks (GANs) simulate malware behavior for model training. Transfer learning and pre-trained models further streamline training and improve accuracy in data-scarce scenarios. Lightweight models are developed for resource-constrained devices, and cloud-based platforms support scalable and centralized detection systems.

Efforts also focus on dataset diversity, explainable AI for trustworthiness, and real-time systems for immediate threat response. Emerging techniques like attention mechanisms and ensemble models continue to refine detection precision and reduce false positives, marking deep learning as a vital tool in combating sophisticated malware.

III. METHODOLOGY

This methodology section we will explain the approach that will guide our research.

Before using the proposed technique, the malware executable files are first converted into images. Then, the proposed fine-tuned Convolutional Neural Network (CNN) model is used to identify and classify malware families from the converted image data. By doing this, researchers learn about the many malware variants and their behaviours. After finishing data processing, section 4.1illustrates the parameters and parameters tuning, which required additional computational capability during the training and testing period. So, a high-configuration computer with high-speed internet takes less computational time, which is related to accuracy. The experiment was conducted by the following configuration of hardware and software.

3.1 Dataset Collection

To detect malware, researchers feed the image data in CNN by neurons (pixels are also used as tensor TF algorithm) exchange and fine-tuning parameters (bias, weights), specific hyper-parameters (number of neurons, kernels/filters, stride), learning rate, minimum batch size, and number of epochs. The model-specific hyper-parameters (number of hidden units, first layer, number of layers), activation function, optimizer, and clusters in K-means). Deep learning requires an enormous numberof datasets for training, but there are a small number of recent image-based malware datasets available, and there are also difficulties that datasets need to be balanced for training and testing. The suggested model worked with a newly created binary-to-image dataset that included 36,551 samples from 34 different types of malware ,like adware ,botnets, spyware, malspam , and exploits .This process looks into the original PE byte stream from an image that has been resized from 32 to 128 in both grayscale and RGB formats. It uses the Nearest Neighbor Interpolation algorithm for the image processing and then converts it into a 1024 byte vector. The purposes of the new dataset are malware identification.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

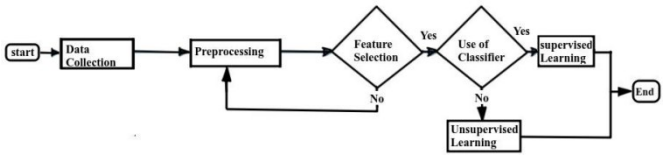

Fig. 1. Process flow diagram of deep learning-based malware identification

3.2Data Conversion Techniques

Using the RegEx pattern finding in Python 3.7, convert raw PE to RGB, and grey-scale image files,saved in .png format, are shown in Figure 3 below. A lot of simulations were carried out to find the best percentages for the training and testing sets to achieve effective malware detection with great accuracy. The findings revealed the ratio (80:20) of the image dataset achieved the suggested and superior efficiency compared with the other ratio for fine-tuned CNN algorithms.

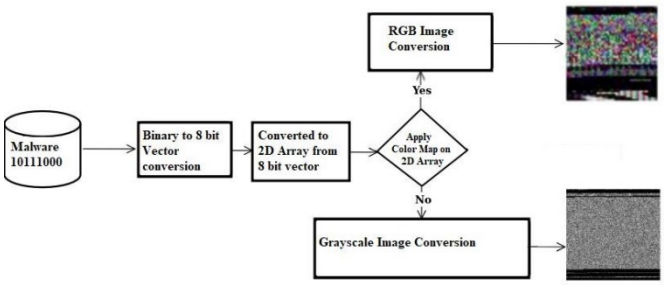

Fig. 2. PE binary file converted to a grayscale and RGB image data set

3.3Normalization

Feeding the dataset into a DL model requires normalization because various dimensions are produced from various PE files, which are challenging to categorize. The images were re-sized to 128×128 pixels to solve this problem.

3.4Identification of Attacks using CNN

CNN can recognize malware by using characteristics similar to how human eyes detect highlights to distinguish between different objects. To construct a customized multilayer and binary CNN model with fine-tuning from the traditional convolution layer to the fully connected layer, weight, pooling, dropout, and kernel are required to adjust properly. The most popular model form in Python programming is the sequential type, and Keras is the simplest method for creating a layer-by-layer CNN model. To build a multiple-layer CNN, with an input layer size of 512 and a kernel capacity of 34×73024, CNN started modelling. An initial convolution layer is present, with a batch normalization of 128 .

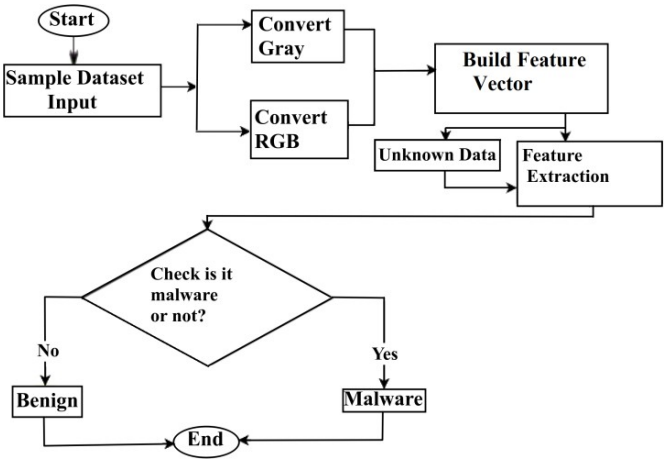

Fig. 3. Flow diagram of malware detection using CNN.

IV. IMPLEMENTATION DETAILS

The implementation of malware detection using deep learning involves a comprehensive process to create an effective system capable of identifying and mitigating threats. It begins with the collection of extensive datasets, including both malware and benign samples, which are preprocessed to remove noise and standardize formats for analysis. Features such as opcode sequences, API call behaviors, or binary file patterns are extracted to represent malware characteristics. Various deep learning models, like Convolutional Neural Networks (CNNs) for image-based malware analysis or Recurrent Neural Networks (RNNs) for sequential data, are trained using these features.

The training process involves supervised learning with labeled datasets, and advanced techniques like adversarial training are incorporated to strengthen resistance against evasion tactics, such as malware obfuscation. Model evaluation uses metrics like precision, recall, and F1-score to ensure high accuracy and robustness. Once validated, the model is deployed in real-time systems for endpoint

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

protection or intrusion detection. Continuous updating and retraining of the model with new malware samples ensure adaptability to emerging threats. Additionally, integrating explainable AI tools helps interpret model decisions, enhancing trust and aiding in precise threat mitigation. This approach ensures a scalable, accurate, and adaptive solution to modern cybersecurity challenges.

To implement the model, upload a customized dataset to Google Drive and connect with Google Colab notebooks to access the dataset. The Python 3.7 language is used for coding to visualize and normalize data, and the DL model applies for training, testing, and validation over days. The confusion matrices are used to evaluate the accuracy of the models. The epochs for the dataset were set to tenfor testing purposes after fine-tuning the hyperparameters. Initially, the proposed model dataset hadan unbalanced distribution of malware and benign PE files despite the author's use of horizontal flipaugmentation to balance the data set.

IV. PROPOSED SYSTEM

The proposed system for malware detection using deep learning is designed to address the limitations of traditional detection methods and improve accuracy, adaptability, and efficiency. The system leverages advanced deep learning techniques, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to analyze malware characteristics effectively.

The first step involves the collection of diverse datasets comprising both begin and malicious files. We clean up these datasets to get rid of any unnecessary information and make sure everything is consistent. Feature extraction plays a vital role in the system, capturing critical patterns such as opcode sequences, network behavior, API calls, and binary file structure. The extracted features are input into the deep learning model, which is trained using labeled data.

During training, techniques like data augmentation, adversarial training, and cross-validation are employed to enhance model robustness against evasion strategies like obfuscation and polymorphism. The proposed system also includes explainable AI components to provide transparency in decision-making, which is essential for building trust in automated cybersecurity solutions.

V. LITERATURE SURVEY

Paper 1:

The "Enhancing Malicious URL Detection: A Novel Framework Leveraging Priority Coefficient and Feature Evaluation" introduces an advanced method for identifying malicious URLs by combining priority coefficients with a feature evaluation mechanism. This framework enhances traditional detection techniques by assigning different importance levels to URL features, allowing for more accurate threat classification. It uses a variety of dynamic features, such as domain, URL length, and special character analysis, to evaluate potential risks. The framework improves detection performance by prioritizing significant features that contribute most to the malicious nature of the URL. By integrating machine learning models with a systematic feature-ranking approach, it reduces false positives and enhances efficiency in identifying new threats. This methodology is especially effective in handling complex and obfuscated URLs. The proposed system demonstrates higher accuracy and reliability compared to traditional detection techniques.

Paper 2: The "Multi-Modal Features Representation-Based Convolutional Neural Network Model for Malicious Website Detection" presents a novel approach for detecting malicious websites using a convolutional neural network (CNN) model that integrates multi-modal features. The model combines various types of data, such as HTML content, URL structure, and domain features, to enhance the accuracy of malicious website detection. By using these diverse input features, the model can capture a more comprehensive representation of website characteristics, leading to better classification. The CNN model processes these features through multiple layers, learning from both visual and textual cues to identify malicious patterns. This approach outperforms traditional methods by improving detection rates and reducing false positives. Additionally, the model is designed to be scalable and adaptable to new types of malicious website threats, making it a robust solution for cybersecurity applications.

Paper 3: A "Secure QR Code Scanner According to a Novel Malicious URL Detection Framework" integrates a novel framework for

detecting and preventing the scanning of malicious QR codes. This framework leverages advanced algorithms to analyze QR code content for potential threats, such as malicious links or scripts. By scanning QR codes in real-time and applying machine learning techniques, the system can quickly identify and block harmful QR codes, protecting users from cyber threats. This approach enhances the security of mobile applications that utilize QR code scanning, making it a critical tool in preventing malicious activities.

VII. CONCLUSION AND FUTURE WORK

The researcher developed and validated a novel strategy to automatically detect and segment binary PE to an image-based malware dataset using a fine-tuned deep-learning model. To achieve this goal, the researcher proposed a novel malware dataset that converted to grayscale and RGB image datasets. Then, it applies customization techniques by fine-tuned DL model where all hyper-parameters are not only tuned using the default tuning function of tensor-flow but also manually tuned the parameters to achieve acceptable accuracy for malware detection. The CNN-based deep learning model was trained without the need for manually sketched PEs before being fine-tuned, which is one of its distinctive features. Now, this proposed model can be used as a transfer learning without manually tuning any parameters for malware detection purposes because the RGB malwaredataset and model parameters problems are almost solved. The proposed fine-tuned CNN model and dataset achieved an accuracy of 98.7% and a rate of error of 1.3% and performed better than other datasets and models. Accuracy can be improved by using a balanced dataset and using more variant attention layers for selecting the main feature, but it requires more computation power, which increases the cost. It is observed that in a balanced dataset of high-resolution RGB three-channel images, horizontal flip augmentation can extract more features than in a one-channel grayscale image dataset. Fitting the model with RGB and grayscale images requires fine-tuned hyper- parameters for malware identification. In conclusion, the application of deep learning to the detection of malicious applications represents a significant advancement in cybersecurity. Unlike traditional methods, which often struggle to evolving of cybercriminals, deep learning models are capable of automatically identifying complex patterns and behaviors that indicate malicious intent.

Furthermore, the integration of deep learning into malware detection systems has the potential to significantly enhance the security of applications across various platforms, safeguarding users and their data from malicious actors.

REFERENCES

[1] Gulatas, Ibrahim, H. Hakan Kilinc, A. Halim Zaim, and M. Ali Aydin. "Malware threat on edge/fog computing environments from Internet of things devices perspective." IEEE Access 11 ,2023

[2] Islam, Chowdhury Sajadul, and Md Sarwar Hossain Mollah. "A novel idea of malaria identification using Convolutional Neural Networks (CNN)." Sep, 2018 .

[3] Smmarwar, Santosh K., Govind P. Gupta, and Sanjay Kumar. "Deep malware detection framework for IoT-based smart application"Aug,2021.

[4] Asam, Muhammad, Shaik Javeed Hussain, Mohammed Mohatram, Saddam Hussain Khan, Tauseef Jamal, Amad
Zafar, Asifullah Khan, Muhammad Umair Ali, and Umme Zahoora. "Detection of exceptional malware variants using deep boosted feature spaces and machine learning.", Feb. 2022.

[5] Conti, Mauro, Shubham Khandhar, and P. Vinod. "A few-shot malware classification approach for unknown family recognition malware feature."Dec,2020.

[6] Akhtar, Muhammad Shoaib, and Tao Feng. "Evaluation of machine learning algorithms for malware detection."ITM Web Conf., vol. 35, 2020, p. 2001.

[7] A. Muhammad, Q. Zhou, G. Beydoun, D. Xu, and J. Shen, "A Review on Occluded Object Detection and Deep Learning Based Approach in Medical Imaging-Related Research." in Proc. IEEE 20th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD), May 2016, pp. 421–426.

[8] Akhtar, Muhammad Shoaib, and Tao Feng. "Malware analysis and detection using machine learning algorithms."Jan,2022.

[9] L. Meng, W. Zhang, Y. Chu, and M. Zhang, "Artificial intelligence-based malware detection, analysis, and mitigation.",IEEE Trans. Learn. Technol., vol. 14, no. 1, pp. 122–128, Feb. 2021.

[10] A. A. Mubarak, H. Cao, and W. Zhang, "A review of android malware detection based on

machine learning.' Interact. Learn. Environ., vol. 30, no. 8, pp. 1414–1433, Jul. 2022.
[11] Naeem, Hamad, Farhan Ullah, Muhammad Rashid Naeem, Shehzad Khalid, Danish Vasan, Sohail Jabbar, and Saqib Saeed. "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model."Comput. Educ., vol. 143, Jan. 2020, Art. no. 103676.
[12] B. K. Bhardwaj and S. Pal, "Image-based malware classification using VGG19 network and spatial convolutional attention."2012
[13] Y. Zhao, Q. Xu, M. Chen, and G. M. Weiss, "A two - stage deep learning framework for image - based Android malware detection and variant classification."in Proc. Int. Educ. Data Mining Soc., Jul. 2020, pp. 1–9.
[14] G. Su-Hui, B. Cheng-Jie, and W. Quan, ''Hadoop-based college student behavior warning decision system,'' in Proc. IEEE 3rd Int. Conf. Big Data Anal. (ICBDA), Mar. 2018, pp. 217–221.
[15] A. Akram, C. Fu, Y. Li, M. Y. Javed, R. Lin, Y. Jiang, and Y. Tang, Binbusayyis, Adel, and Thavavel Vaiyapuri. "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM."IEEE Access, vol. 7, pp. 102487–102498, 2019.

[16] N. Hidayat, R. Wardoyo, A. Sn, and H. Dwi, . "Network intrusion detection model using one-class support vector machine."Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 3, pp. 638–648, 2020.
[17] N. T. Nghe, P. Janecek, and P. Haddawy, . "Network intrusion detection model using one-class support vector machine."in Proc. 37th Annu. Frontiers Educ. Conf. Global Eng., Knowl. Without Borders, Opportunities Without Passports, Oct. 2007, p. 7.
[18]Min, Byeongjun, Jihoon Yoo, Sangsoo Kim, Dongil Shin, and Dongkyoo Shin. "Network anomaly detection using memory-augmented deep autoencoder." IEEE Access 9 2021.
[19] guyen, Quoc Thong, Kim Phuc Tran, Philippe Castagliola, Truong Thu Huong, Minh Kha Nguyen, and Salim Lardjane. "Nested one-class support vector machines for network intrusion detection." Appl. Sci., vol. 10, no. 3, p. 1042, Feb. 2020.
[20] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, ''Analyzing undergraduate students' performance using educational data mining,'' Comput. Educ., vol. 113, pp. 177–194, Oct. 2017.
[21] Khraisat, Ansam, Iqbal Gondal, Peter Vamplew, Joarder Kamruzzaman, and Ammar Alazab. "Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine."Smart Learn. Environ., vol. 9, no. 1, p. 11, 2022.
[22] Qi, Ruobin, Craig Rasband, Jun Zheng, and Raul Longoria. "Detecting cyber attacks in smart grids using semisupervised anomaly detection and deep representation learning." Artif. Intell., vol. 3, Jan. 2022, Art. no. 100066.
[23] Kamboj, Akshit, Priyanshu Kumar, Amit Kumar Bairwa, and Sandeep Joshi. "Detection of malware in downloaded files using various machine learning models." IEEE Access, vol. 7, pp. 19550–19563, 2019.
[24] Venkatraman, Sitalakshmi, Mamoun Alazab, and R. Vinayakumar. "A hybrid deep learning image-based analysis for effective malware detection." Journal of Information Security and Applications 47 (2019): 377-389.IEEE Access, vol. 8, pp. 203827–203844, 2020.
[25] Zhu, Huijuan, Huahui Wei, Liangmin Wang, Zhicheng Xu, and Victor S. Sheng. "An effective end-to-end android malware detection method." Expert System With Applications 218 (2023): 119593.
[26] Chaganti, Rajasekhar, Vinayakumar Ravi, and Tuan D. Pham. "Deep learning based cross architecture internet of things malware detection and classification." Computers & Security 120 (2022): 102779.