International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

A Hybrid Cybersecurity Model for Malware Detection Using Convolutional Neural Networks and Random Forest Classifiers

Mrs. P. Jenifer*, Ms. G. Jenisha Samlin **
*(Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: jenifer@francisxavier.ac.in)
** (Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: samlinjenii@gmail.com)

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

Abstract:
Malware detection is the process of identifying malicious software such as viruses, worms, trojans, spyware, and ransomware that harm devices or steal sensitive information. It works by examining files, programs, system behavior, or network activity to spot unusual or unsafe patterns. The goal of malware detection is to prevent unauthorized access, data loss, and system damage by quickly identifying harmful activity and blocking its execution. A major traditional issue in malware detection is poor adaptability. To address this issue, this project proposes a hybrid cybersecurity model for malware detection using convolutional neural networks (CNNs) and random forest classifiers. The system combines deep learning and machine learning techniques to improve detection accuracy and efficiency.

Keywords —***Malware Detection, CNN, Random Forest, Cybersecurity, Machine Learning, Deep Learning***

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

I. INTRODUCTION

In today's digital world, cybersecurity has become a critical concern due to the rapid increase in cyber threats. Malware is one of the most dangerous threats that can damage systems, steal data, and disrupt services. Traditional malware detection techniques, such as signature-based methods, are not effective against new and unknown threats.

Machine learning and deep learning techniques provide advanced solutions for detecting malware by analyzing patterns and behaviors. This project introduces a hybrid model that combines Convolutional Neural Networks (CNN) and Random Forest classifiers to improve detection performance.

II. OBJECTIVES

The primary objective of this project is to design and develop an efficient and intelligent malware detection system using a hybrid approach that combines deep learning and machine learning techniques. The system aims to enhance detection accuracy, adaptability, and reliability in identifying malicious software.

A. Development of Hybrid Model:
To design and implement a hybrid cybersecurity model that integrates Convolutional Neural Networks (CNN) for feature extraction and Random Forest for classification. This combination aims to leverage the strengths of both deep learning and ensemble learning techniques.

B. Improvement in Detection Accuracy:
To enhance the accuracy of malware detection while minimizing false positives and false negatives. The system should be capable of identifying both known and unknown malware effectively.

C. Deep Feature Extraction:
To utilize CNN for automatic extraction of high-

 ©IJSRED: All Rights are Reserved

level and meaningful features from raw input data, reducing the need for manual feature engineering.

D. Efficient Classification Mechanism:
To employ Random Forest algorithm for robust and efficient classification of malware based on extracted features, ensuring better generalization and reduced overfitting.

E. Performance Evaluation:
To evaluate the system performance using standard evaluation metrics such as accuracy, precision, recall, and F1-score, ensuring the reliability of the proposed model.

III. MODULES AND ALGORITHMS

The proposed system is divided into several modules to ensure structured processing and efficient malware detection. The design of modules follows a systematic approach similar to the reference journal

1) A. Modules

1) Data Collection Module:
This module is responsible for collecting datasets containing both benign and malicious software samples. The dataset forms the foundation of the system and plays a crucial role in training and testing the model.

2) Data Preprocessing Module:
In this module, the collected data undergoes preprocessing steps such as handling missing values, normalization, and feature scaling. Dimensionality reduction techniques may also be applied to improve efficiency and reduce computational complexity.

3) Feature Extraction Module (CNN):
This module utilizes Convolutional Neural Networks to automatically extract deep and meaningful features from the input data. CNN helps in identifying hidden patterns and relationships within the data, which are not easily detectable using traditional methods.

4) Classification Module (Random Forest):
The extracted features are passed to the Random Forest classifier, which uses multiple decision trees to classify the data into malicious or benign categories. This improves accuracy and reduces the risk of overfitting.

5) Evaluation Module:
This module evaluates the performance of the model using metrics such as accuracy, precision, recall, and F1-score. These metrics help in analyzing the effectiveness of the system.

2) B. Algorithms

1) Convolutional Neural Network (CNN):
CNN is a deep learning algorithm widely used for feature extraction. It consists of multiple layers such as convolutional layers, pooling layers, and fully connected layers. These layers work together to extract hierarchical features from input data. CNN automatically learns patterns, making it highly effective for malware detection.

2) Random Forest Algorithm:
Random Forest is an ensemble learning technique that constructs multiple decision trees during training and outputs the majority class as the final prediction. It improves classification accuracy, handles large datasets efficiently, and reduces overfitting by combining multiple models.

3) C. Functional Modules

1) Feature Learning Mechanism:
The CNN model learns complex patterns and representations from the dataset automatically, enabling the system to detect malware based on deep features.

2) Classification Logic:
The Random Forest classifier processes the extracted features and classifies the input as either malicious or benign with high accuracy.

 ©IJSRED: All Rights are Reserved

IV. METHODOLOGY

The proposed malware detection system follows a systematic and structured methodology to ensure accurate and efficient identification of malicious software. The overall workflow is designed based on advanced machine learning and deep learning techniques, similar to the structured approach presented in the reference journal

A. Data Collection

The first step in the proposed system is data collection, which plays a crucial role in determining the effectiveness of the model. The dataset consists of both benign and malicious software samples collected from reliable sources. These samples may include executable files, network traffic data, or system activity logs.

The collected dataset must be diverse and representative of real-world scenarios to ensure that the model can generalize well. A balanced dataset is maintained to avoid bias in classification. Proper labeling is also performed to distinguish between malicious and non-malicious samples.

B. Data Preprocessing

In this stage, the collected raw data is processed to make it suitable for training the model. Data preprocessing involves several steps such as data cleaning, handling missing values, normalization, and feature scaling.

Noise and irrelevant data are removed to improve the quality of the dataset. In addition, dimensionality reduction techniques may be applied to reduce computational complexity while preserving important features. This step ensures that the data is consistent, accurate, and ready for feature extraction.

C. Feature Extraction using CNN

Feature extraction is performed using Convolutional Neural Networks (CNN), which are highly effective in identifying patterns within complex data. The preprocessed data is fed into the CNN model, where multiple layers such as convolutional layers, pooling layers, and fully connected layers are used to extract deep features.

CNN automatically learns hierarchical representations from the input data, capturing both low-level and high-level features. This eliminates the need for manual feature engineering and improves the model's ability to detect hidden patterns associated with malware.

D. Model Training and Testing

After feature extraction, the dataset is divided into training and testing sets. The training dataset is used to train the model, while the testing dataset is used to evaluate its performance.

During training, the CNN model learns feature representations, and the Random Forest classifier is trained using these extracted features. Proper validation techniques are applied to avoid overfitting and ensure that the model performs well on unseen data.

E. Classification using Random Forest

In this stage, the extracted features are provided as input to the Random Forest classifier. Random Forest is an ensemble learning algorithm that constructs multiple decision trees and combines their outputs to produce a final prediction.

Each decision tree in the forest makes an independent prediction, and the final classification is determined based on majority voting. This approach improves accuracy, reduces variance, and enhances the robustness of the system. The classifier categorizes the input data as either malicious or benign.

F. Performance Evaluation

 ©IJSRED: All Rights are Reserved

The final stage involves evaluating the performance of the proposed system using standard evaluation metrics. These include accuracy, precision, recall, and F1-score.

- **Accuracy** measures the overall correctness of the model.
- **Precision** indicates how many predicted malware instances are actually correct.
- **Recall** measures the ability of the model to detect actual malware.
- **F1-score** provides a balance between precision and recall.

These metrics help in assessing the effectiveness and reliability of the system. A higher value in these metrics indicates better performance of the malware detection model.

VI. EXISTING SYSTEM

Traditional malware detection systems have been widely used in cybersecurity; however, they suffer from several limitations that reduce their effectiveness against modern and evolving threats. These systems primarily rely on predefined rules and patterns, which are not sufficient to handle sophisticated malware attacks. Similar limitations are also discussed in the reference journal .

A. Signature-Based Detection System

Signature-based detection is one of the most commonly used techniques in traditional antivirus systems. It works by comparing files against a database of known malware signatures. While this method is effective for detecting previously identified threats, it fails to detect new or unknown malware (zero-day attacks). Since attackers continuously modify malware to avoid detection, signature-based systems become outdated quickly.

B. Heuristic-Based Detection System

Heuristic-based detection attempts to identify malware by analyzing patterns, behaviors, and suspicious activities. Although this method improves detection compared to signature-based approaches, it often generates inaccurate results. It may incorrectly classify legitimate files as malicious or fail to detect cleverly disguised malware, leading to reduced reliability.

C. Limited Adaptability

Traditional systems lack adaptability to new and evolving malware techniques. They depend heavily on predefined rules and manual updates, which makes them slow in responding to emerging threats. As cyber attackers develop advanced techniques, these systems struggle to keep up with dynamic changes in malware behavior.

4) D. High False Positive Rate

Another major limitation of existing systems is the high rate of false positives. Benign files are often misclassified as malicious, which can disrupt normal system operations and reduce user trust. This issue arises due to the lack of intelligent learning mechanisms in traditional approaches.

5) E. Lack of Intelligent Feature Learning

Conventional systems do not utilize automated feature learning techniques. Instead, they rely on manually engineered features, which may not capture complex patterns in malware data. This limitation reduces detection accuracy and overall system efficiency.

6) Summary of Existing System Limitations

Due to the above limitations, traditional malware detection systems are not suitable for modern cybersecurity environments. There is a need for advanced intelligent systems that can automatically learn patterns, adapt to new threats, and provide accurate detection.

VII. PROPOSED SYSTEM

 ©IJSRED: All Rights are Reserved

To overcome the limitations of existing systems, a hybrid cybersecurity model is proposed that combines Convolutional Neural Networks (CNN) and Random Forest classifiers. This system leverages the strengths of both deep learning and machine learning techniques to achieve efficient and accurate malware detection.

A. Hybrid Detection Approach

The proposed system integrates CNN for feature extraction and Random Forest for classification. CNN automatically learns deep and meaningful features from input data, while Random Forest uses these features to perform accurate classification. This hybrid approach enhances overall system performance and ensures better detection capability.

B. Improved Detection Accuracy

By combining deep learning and ensemble learning methods, the proposed system significantly improves detection accuracy. CNN captures complex patterns in the data, and Random Forest ensures robust decision-making. This results in more reliable classification compared to traditional methods.

C. Reduction in False Positives

The proposed system minimizes false positive rates by using advanced learning techniques. The combination of CNN and Random Forest helps in distinguishing between benign and malicious samples more effectively, reducing incorrect classifications.

D. Enhanced Adaptability

Unlike traditional systems, the proposed model can adapt to new and unknown malware. CNN enables automatic feature learning, allowing the system to identify previously unseen patterns. This makes the system suitable for handling evolving cyber threats.

E. Automated Feature Extraction

The system eliminates the need for manual feature engineering by using CNN for automatic feature extraction. This improves efficiency and allows the model to focus on relevant patterns in the data.

F. Scalable and Efficient Architecture

The proposed system is scalable and can be extended to handle large datasets and real-time applications. It provides an efficient framework for malware detection in modern cybersecurity environments.

VIII. OUTPUT

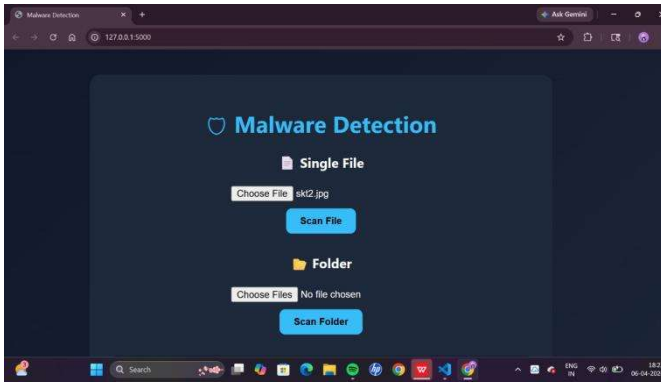
a) User Interface for File and Folder Scanning

The first screen shows the main interface of the malware detection system, where the user can upload either a single file or multiple files (folder) for analysis. The “Choose File” option allows selecting one file, and the “Scan File” button is used to start scanning. Similarly, the folder section allows uploading multiple files and scanning them together using the “Scan Folder” button. This interface acts as the input stage where users provide files for malware analysis.

 ©IJSRED: All Rights are Reserved

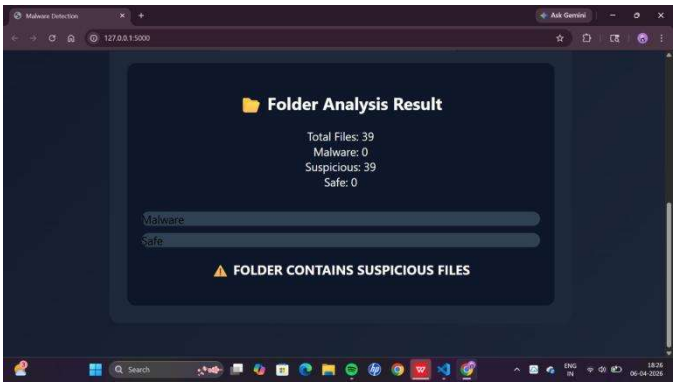
b) Malware Detection Result for a Single File

The second screen displays the result of a scanned file, which is classified as suspicious. The system shows that the file has a 39% probability of being benign and a 61% probability of being malware, indicating a higher chance of being malicious. The confidence level is 61%, meaning the model is moderately sure about its prediction. The file is categorized under medium risk, suggesting that it is not completely safe and should be handled with caution. Additionally, a moderate anomaly in the file structure is detected, and the entropy value of 7.868 indicates high randomness, which is commonly seen in encrypted or obfuscated malicious files.



c)File Selection Before Scanning

The third screen shows an example where a specific file (skt2.jpg) has been selected for scanning. This step represents the process of choosing a file before performing malware detection. It confirms that the system supports different file types and allows users to preview the selected file before initiating the scan.



d) Folder Analysis and Detection Result

The fourth screen presents the analysis result of a folder containing 39 files. The system reports that none of the files are clearly identified as malware, but all 39 files are marked as suspicious, and no files are considered safe. This indicates that every file in the folder has unusual characteristics. As a result, the system concludes that the folder contains suspicious files and recommends further inspection before using them.

IX. CONCLUSIONS

This project presents a hybrid cybersecurity model for malware detection using Convolutional Neural Networks (CNN) and Random Forest classifiers. The proposed system effectively addresses the limitations of traditional malware detection techniques by incorporating advanced machine learning and deep learning approaches.

Furthermore, the modular architecture of the system allows easy scalability and integration with real-world applications. The results indicate that the proposed hybrid model outperforms traditional methods in terms of performance and efficiency.

X . ACKNOWLEDGEMENT

We express our sincere and heartfelt gratitude to our mentor, , Assistant Professor, Department of Computer Science and Engineering, for her invaluable guidance, continuous encouragement, and expert knowledge, which significantly

 ©IJSRED: All Rights are Reserved

contributed to the successful completion of this research work. Her support and insightful suggestions played a crucial role in shaping the quality and direction of this project.

XI . FUTURE ENHANCEMENT

- Implementing real-time malware detection
- Integrating cloud-based security solutions
- Using advanced deep learning models such as LSTM or Transformer
- Expanding the dataset for improved generalization

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[2] L. Nataraj, S. Karthikeyan, G. Jacob, and B. Manjunath, "Malware Images: Visualization and Automatic Classification," Proceedings of the 8th International Symposium on Visualization for Cyber Security, pp. 1–7, 2011.

[3] M. Z. Rafique and J. Caballero, "Firma: Malware Clustering and Network Signature Generation with Mixed Network Behaviors," International Symposium on Research in Attacks, Intrusions and Defenses, pp. 144–163, 2013.

[4] U. Bayer, P. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, Behavior-Based Malware Clustering," NDSS, 2009.

[5] S. Y. Yerima, S. Sezer, and I. Muttik, "High Accuracy Android Malware Detection Using Ensemble Learning," IET Information Security, vol. 9, no. 6, pp. 313–320, 2015.

[6] A. Pektaş and T. Acarman, "Deep Learning for Effective Android Malware Detection Using API Call Graph Embeddings," Soft Computing, vol. 24, pp. 10277–10289, 2020.

[7] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[8] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," OSDI, pp. 265–283, 2016.

[9] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436–444, 2015.

[11] K. Saxe and K. Berlin, "Deep Neural Network Based Malware Detection Using Two Dimensional Binary Program Features," 10th International Conference on Malicious and Unwanted Software (MALWARE), 2015.

[12] J. Z. Kolter and M. A. Maloof, "Learning to Detect Malicious Executables in the Wild," KDD, pp. 470–478, 2004.

 ©IJSRED: All Rights are Reserved