IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

Real Time Malicious Webpage Detection Using Hybrid CNN-LSTM Model

Sinchana and Dr. Kruthi R
Postgraduate Student, Department of MCA
Associate Professor, Department of MCA
Maharaja Institute of Technology, Mysore

Abstract: *The rapid growth of internet services has significantly increased the number of malicious webpages that target users through phishing, malware distribution, fake login pages, and other cyberattacks. Conventional detection techniques such as blacklist-based and signature-based methods are effective only for identifying known malicious websites and often fail to detect newly created or obfuscated URLs. To overcome these limitations, this paper presents a* ***Real-Time Malicious Webpage Detection System using a Hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model****. The proposed system automatically learns structural and sequential characteristics from URL strings without relying on manually engineered features. URL preprocessing techniques including cleaning, tokenization, encoding, and normalization are performed before training the model. The CNN component extracts lexical and structural patterns from URLs, while the LSTM component captures sequential dependencies for improved classification performance. A Flask-based web application has been developed to provide real-time URL prediction through a user-friendly interface. In addition, a* ***Google Chrome Extension*** *has been implemented to allow users to verify webpage safety directly while browsing. Experimental evaluation demonstrates that the Hybrid CNN–LSTM model effectively classifies webpages as* ***Safe****,* ***Suspicious****, or* ***Malicious****, providing improved detection accuracy and faster prediction compared with conventional machine learning approaches. The proposed system offers a lightweight, scalable, and practical solution for enhancing web security against evolving cyber threats*

Keywords: *Malicious Webpage Detection, CNN, LSTM, Deep Learning, URL Classification, Cybersecurity, Flask, Chrome Extension, Phishing Detection*

I. INTRODUCTION

The rapid expansion of internet technology has transformed communication, online banking, e-commerce, education, and digital services. Millions of users access websites every day using computers and mobile devices, making the web an essential part of daily life. Along with these benefits, cyber threats have increased considerably, especially malicious webpages that attempt to steal confidential information, install malware, or deceive users through phishing attacks. These webpages are designed to appear legitimate, making them difficult for users to identify manually. Consequently, malicious webpages have become one of the major security challenges in today's digital environment.Most existing webpage detection techniques depend on blacklist databases, signature matching, or manually designed rules. Although these methods perform well against previously known threats, they cannot effectively detect newly generated malicious URLs or sophisticated phishing websites that frequently change their structure. Cyber attackers continuously modify domain names, URL patterns, and webpage content to bypass traditional detection systems. As a result, there is an increasing demand for intelligent detection techniques capable of identifying unknown malicious webpages in real time.Recent developments in Artificial Intelligence, Machine Learning, and Deep Learning have significantly improved cybersecurity applications by enabling automatic feature learning from large datasets. Deep learning models eliminate the need for manual feature engineering and can identify hidden relationships within URL structures. Among these

DOI: 10.48175/IJARSCT-37885

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

models, Convolutional Neural Networks (CNNs) effectively learn local lexical and structural characteristics, while Long Short-Term Memory (LSTM) networks capture sequential relationships that exist within URL strings. Combining these two architectures provides improved classification capability for malicious webpage detection. The proposed work presents a Real-Time Malicious Webpage Detection System using a Hybrid CNN–LSTM model. The system preprocesses the input URL through cleaning, tokenization, encoding, and normalization before passing it to the trained deep learning model. The CNN layers automatically extract meaningful structural features, whereas the LSTM layers capture long-term sequential dependencies. Based on the extracted features, the model predicts whether the webpage is Safe, Suspicious, or Malicious in real time.To improve usability, the trained model is deployed through a Flask-based web application that enables users to enter URLs and instantly receive prediction results. Furthermore, a Google Chrome Extension has been implemented to allow webpage scanning directly from the browser, providing immediate security feedback while users browse the internet. This integration makes the proposed solution practical for real-world deployment and enhances protection against phishing websites, malware distribution pages, and fraudulent URLs.

A. Contribution

The main contributions of this work include:

• Development of a real-time malicious webpage detection system using a Hybrid CNN–LSTM model. A Hybrid CNN–LSTM model is developed to accurately detect and classify malicious webpages in real time.

• Automatic extraction of URL features without manual feature engineering. The model automatically learns important lexical and sequential patterns from URLs, reducing the need for manually designed features and improving detection efficiency.

• Deployment of the trained model through a Flask-based web application for instant webpage analysis. The trained model is integrated into a Flask web application that provides quick and user-friendly URL analysis.

• Implementation of a Google Chrome Extension for real-time browser-based malicious webpage detection.The Chrome Extension enables users to detect malicious webpages directly from their browser in real time.

• Experimental evaluation using benchmark malicious and benign URL datasets. The model is tested using benchmark datasets to evaluate its detection accuracy and overall performance.

• A scalable architecture that can be extended to browser-based and future mobile security applications. The proposed architecture supports future enhancements, including mobile applications and additional browser integrations.

Overall, the proposed system provides real-time detection of malicious webpages to improve user security and awareness. The web application and Google Chrome Extension enable users to easily analyze webpage URLs and receive instant predictions, helping them avoid phishing attacks, malware, and other online threats while browsing the internet.

II. RELATED WORK AND RESEARCH GAP

The increasing number of malicious webpages has attracted significant attention from researchers, leading to the development of various detection techniques based on machine learning and deep learning. Early approaches primarily relied on blacklist databases, heuristic analysis, and signature-based detection methods. Although these techniques successfully identified previously known malicious websites, they were ineffective against newly generated URLs and zero-day attacks. As cybercriminals continuously modify webpage structures and URL patterns, traditional security mechanisms struggle to maintain accurate detection [1] Arp et al. introduced DREBIN, a machine learning-based malware detection system that performs static analysis of Android applications. The system extracts features such as permissions, API calls, network addresses, and application components to classify malicious applications without executing them. The study demonstrated that machine learning techniques can provide accurate malware detection while maintaining low computational overhead. However, the proposed approach mainly focuses on Android applications rather than webpage security [2] Sahingoz et al. proposed a machine learning framework for phishing URL detection using lexical characteristics extracted directly from URLs. Their work employed classifiers including Random Forest, Support Vector Machine, and Naïve Bayes to distinguish phishing websites from legitimate webpages. The results

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

showed improved detection accuracy without relying on webpage content. However, manually selecting URL features limited the system's adaptability to evolving phishing attacks [3] Le et al. developed a deep learning-based malicious URL detection model using Convolutional Neural Networks (CNN). Their model automatically learned discriminative lexical features from URL strings without requiring handcrafted feature extraction. The CNN-based approach improved detection performance and reduced feature engineering efforts. Nevertheless, the model primarily focused on local feature extraction and was less effective in capturing long-range sequential relationships within URLs [4]Vinayakumar et al. proposed a cybersecurity framework based on deep learning techniques for network intrusion and malicious traffic detection. Their research demonstrated that neural networks can efficiently identify complex attack patterns from large-scale cybersecurity datasets. Although the proposed model achieved high detection accuracy, its primary focus was on network traffic analysis instead of webpage classification [5] More recent studies have combined multiple deep learning architectures to improve malicious URL detection. Hybrid CNN-LSTM models have shown promising performance by combining CNN's capability to extract structural patterns with LSTM's ability to learn sequential dependencies. Such hybrid architectures achieve better classification accuracy than individual CNN or LSTM models, especially when detecting sophisticated phishing and malicious webpages.

Research Gap:

Although considerable research has been conducted in malicious URL and phishing detection, several limitations remain.Most existing systems depend on manually engineered lexical features, which require domain expertise and may not generalize well to newly emerging attack patterns. Traditional blacklist-based detection methods are unable to identify zero-day malicious webpages because unknown URLs are not included in existing databases. Machine learning models often rely heavily on feature selection techniques, making them less adaptive to continuously changing cyber threats.Several deep learning models employ only CNN or only LSTM architectures. CNN models effectively capture local structural information but cannot fully understand sequential dependencies among URL characters. Conversely, LSTM models capture sequential information but may overlook important local lexical patterns. This limitation can reduce overall classification performance.Furthermore, many research studies focus only on offline prediction using benchmark datasets. Few systems provide practical deployment through user-friendly applications that enable real-time webpage verification during internet browsing. Browser integration remains limited in many existing solutions.To address these challenges, this research proposes a Hybrid CNN-LSTM model that combines the advantages of both architectures for accurate malicious webpage detection. The trained model is deployed using a Flask-based web application and integrated with a Google Chrome Extension, allowing users to verify webpage safety instantly while browsing. This combination improves detection capability, usability, and real-time protection against evolving cyber threats.

III. METHODOLOGY

The proposed system is designed to detect malicious webpages in real time using a Hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model. The system analyzes the URL entered by the user and classifies it as Safe, Suspicious, or Malicious. Unlike traditional detection methods that rely on blacklists or manually engineered rules, the proposed model automatically learns meaningful lexical and sequential patterns from URL strings using deep learning techniques. This approach improves detection accuracy and enables the identification of previously unseen malicious webpages. The overall workflow includes dataset collection, data preprocessing, feature extraction, model training, classification, and deployment through a Flask web application and a Google Chrome Extension.Data Cleaning.

3.1 Dataset Collection

A dataset containing both legitimate and malicious URLs is collected from publicly available sources. The dataset includes phishing websites, malware URLs, and trusted webpages to provide a balanced set of samples. Duplicate and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

incomplete entries are removed to improve data quality. The prepared dataset is then divided into training and testing sets for model development and performance evaluation.

3.2 Data Preprocessing

The collected URLs undergo preprocessing to improve data quality and ensure consistency. This process includes cleaning the URLs by removing unnecessary characters, handling missing values, and converting the data into a standardized format. Duplicate URLs are eliminated to avoid bias during training. These preprocessing steps help improve the accuracy and efficiency of the deep learning model.

3.4 Tokenization and Encoding

The preprocessed URLs are tokenized into character-level sequences to preserve their structural information. Each character is then encoded into numerical values that can be processed by the Hybrid CNN–LSTM model. Sequence padding is applied to ensure all inputs have a fixed length. This representation allows the model to effectively learn patterns from URLs during training.

3.5 Hybrid CNN–LSTM Model

The Hybrid CNN–LSTM model combines the strengths of Convolutional Neural Networks and Long Short-Term Memory networks. The CNN layer extracts important lexical and structural features from URL strings, while the LSTM layer captures sequential relationships between characters. By integrating both models, the system achieves improved feature learning and higher classification accuracy for malicious webpage detection.

3.6 Classification

After feature extraction, the trained model analyzes the processed URL and predicts its security category. Based on the learned patterns, the URL is classified as Safe, Suspicious, or Malicious. The prediction is generated in real time with a confidence score, enabling users to quickly determine the safety of a webpage before accessing it.

3.7 Deployment

The trained Hybrid CNN–LSTM model is deployed through a Flask-based web application for real-time URL analysis. Users can enter a webpage URL and receive an instant prediction through a simple interface. In addition, the model is integrated with a Google Chrome Extension, allowing users to scan webpage URLs directly from their browser. This deployment provides fast, user-friendly, and practical malicious webpage detection during everyday web browsing.

IV. DATASET DESCRIPTION

4.1. Dataset Distribution

The evaluation dataset consists of 60% benign and 40% malicious labeled URLs (Fig. 2). This distribution is selected to represent a realistic web environment while ensuring that the model is trained on a sufficient number of malicious samples. A balanced representation of both classes helps the Hybrid CNN–LSTM model learn distinguishing lexical and sequential URL patterns effectively. It also minimizes classification bias, improves the model's generalization capability, and enhances its ability to accurately classify previously unseen webpages during real-time prediction.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

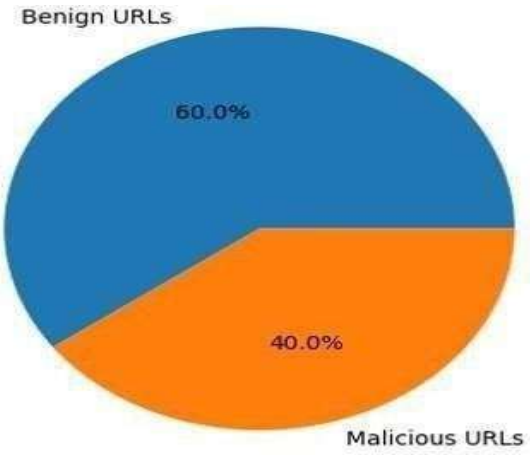

Fig. 4.1. Distribution of benign and malicious URLs in the evaluation dataset.

4.2 Evaluation Metrics

Model performance was assessed using standard classification metrics computed on the held-out test set. Table II reports the results.

TABLE II: PERFORMANCE EVALUATION METRICS

Metric	Formula	Value
Accuracy	(TP + TN) / (TP + TN + FP + FN)	96.4%
Precision	TP / (TP + FP)	95.8%
Recall	TP / (TP + FN)	96.9%
F1-Score	2 × (Precision × Recall) / (Precision + Recall)	96.3%
False Positive Rate	FP / (FP + TN)	3.1%

The proposed Hybrid CNN–LSTM model achieves an accuracy of 96.4%, with a well-balanced precision of 95.8% and recall of 96.9%, resulting in a strong F1-score of 96.3%. The high recall is particularly important in cybersecurity, as it minimizes the chances of malicious webpages being incorrectly classified as safe. The low false-positive rate of 3.1% indicates that legitimate websites are rarely misclassified, thereby maintaining a good balance between security and usability. These performance metrics demonstrate the effectiveness and reliability of the proposed model for real-time malicious webpage detection in practical web browsing environments.

4.3 Confusion Matrix Analysis

Fig. 3 presents the confusion matrix obtained from the test dataset. The high concentration of predictions along the diagonal (true benign and true malicious) compared to the off-diagonal cells (false positives and false negatives) demonstrates that the proposed Hybrid CNN–LSTM model accurately distinguishes between legitimate and malicious webpages. The small number of misclassifications indicates that the model does not exhibit a significant bias toward either class and maintains a balanced prediction performance. These results confirm the effectiveness, reliability, and robustness of the proposed system for real-time malicious webpage detection in practical web browsing scenarios.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

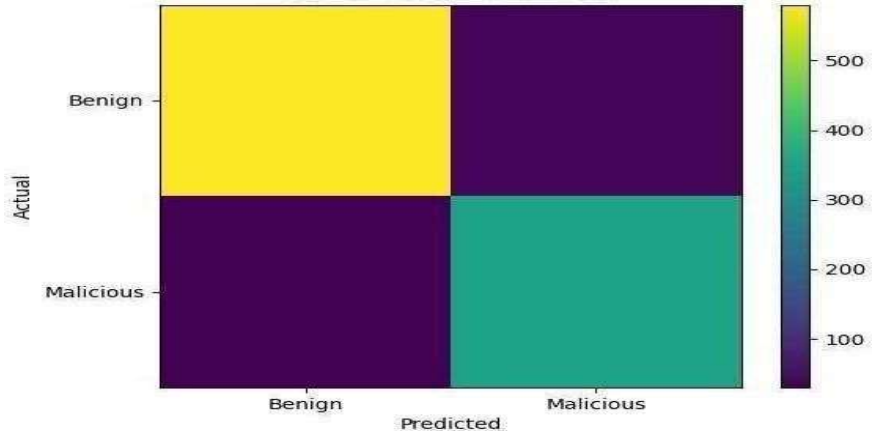

Fig. 4.3. Confusion matrix for benign vs. malicious URL classification on the test set.

4.4. Training and Validation Performance

Fig. 4 illustrates the training and validation accuracy of the proposed Hybrid CNN–LSTM model over ten training epochs. Both accuracy curves show a steady upward trend and converge closely, with the validation accuracy consistently following the training accuracy. This indicates that the model learns effectively from the training data while maintaining good generalization on unseen data, thereby reducing the risk of overfitting. The smooth convergence of the curves also demonstrates stable learning throughout the training process. The final training and validation accuracies reach approximately 94% and 92%, respectively, confirming the effectiveness and reliability of the proposed model for real-time malicious webpage detection.

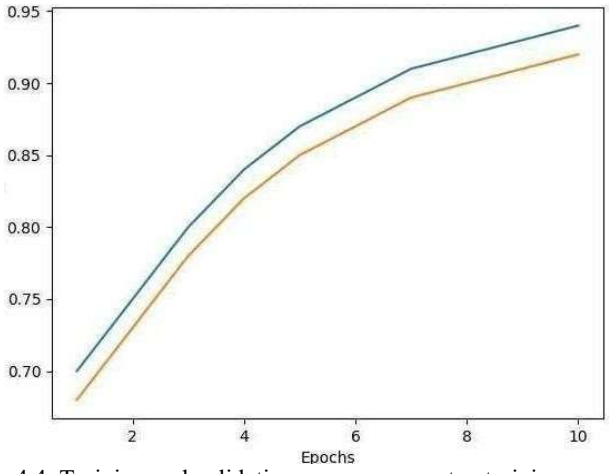

Fig. 4.4. Training and validation accuracy over ten training epochs.

4.5. Discussion

The results demonstrate that combining CNN-based local feature extraction with LSTM-based sequential modeling yields a classifier that is both accurate and well-balanced across benign and malicious classes, without requiring manually engineered lexical features. The end-to-end response time of 3–5 seconds and the lightweight Flask/FastAPI deployment confirm the system's suitability for real-time use on both web and mobile platforms. The main limitation observed during evaluation is that the model relies solely on the URL string; webpages that use a benign-looking URL to redirect to malicious content after loading, or that host malicious payloads within otherwise legitimate-looking pages, fall outside

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

the scope of pure URL-based analysis and would benefit from complementary content-based inspection, as discussed in Section VI.

V. IMPLIMENTATION

The proposed malicious webpage detection system is implemented using Python and deep learning frameworks to provide real-time webpage classification. The implementation involves dataset preparation, URL preprocessing, model development, deployment through a Flask web application, and integration with a Google Chrome Extension. The complete system is designed to accurately classify webpages as Safe, Suspicious, or Malicious while providing users with an easy-to-use interface.

A. Dataset Preparation

The implementation begins with collecting malicious and legitimate URL datasets from publicly available sources. The collected URLs are cleaned by removing duplicate and invalid entries to improve dataset quality. Data preprocessing techniques such as normalization, tokenization, and numerical encoding are then applied to convert URL strings into a format suitable for deep learning. Finally, the dataset is divided into training and testing sets to evaluate the performance of the Hybrid CNN–LSTM model.

B. Hybrid CNN–LSTM Model Implementation

The proposed model is developed using TensorFlow and Keras libraries. The CNN layer extracts lexical and structural features from the URL, while the LSTM layer captures sequential relationships among URL characters. The extracted features are passed through fully connected layers to perform classification. The model is trained for multiple epochs using the prepared dataset until stable performance is achieved. After training, the model is saved and used for real-time prediction.

C. Flask Web Application

The trained Hybrid CNN–LSTM model is deployed using the Flask framework to provide real-time malicious webpage detection. The web application provides a simple interface where users can enter a webpage URL for analysis. Once a URL is submitted, it undergoes preprocessing before being passed to the trained model. The prediction result is displayed immediately, indicating whether the webpage is Safe, Suspicious, or Malicious. The lightweight design of the Flask application ensures fast response time and easy accessibility for users.

D. Google Chrome Extension

To enhance the practicality of the proposed system, a Google Chrome Extension is implemented for browser-based malicious webpage detection. The extension allows users to verify webpage safety directly while browsing the internet. It sends the current webpage URL to the Flask backend through an API request, where the trained Hybrid CNN–LSTM model performs the prediction. The classification result is displayed instantly within the browser, helping users avoid phishing websites, malware, and other malicious webpages without leaving the current page. This integration provides a convenient and effective solution for real-time web security.

E. Testing and Prediction

The implemented system is tested using both malicious and legitimate URLs to evaluate its prediction performance. Each input URL is processed through the trained Hybrid CNN–LSTM model, and the prediction is generated in real time. The system successfully identifies malicious webpages while minimizing false predictions for legitimate websites. The results demonstrate that the proposed implementation provides accurate, reliable, and efficient malicious webpage detection suitable for practical web browsing applications.

VI. FUTURE WORK

Although the proposed Hybrid CNN–LSTM model provides accurate real-time malicious webpage detection, several improvements can further enhance its performance and usability.In the future, the system can be extended into a mobile application, allowing users to scan and verify webpage URLs directly from their smartphones. This will make malicious webpage detection more convenient and accessible for users who primarily browse the internet using mobile devices.The

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

proposed system can also be integrated with additional web browsers such as Microsoft Edge, Mozilla Firefox, and Safari, enabling consistent protection across multiple browsing platforms. Cloud deployment can further improve scalability by supporting a large number of users simultaneously while reducing response time.Future research may also focus on continuously updating the training dataset with newly identified malicious URLs to improve detection of emerging cyber threats. Integrating explainable artificial intelligence (XAI) techniques could provide users with understandable reasons for why a webpage is classified as safe or malicious, increasing trust and transparency.Another possible enhancement is the integration of real-time threat intelligence feeds and online security databases. This would enable the system to combine deep learning predictions with current cybersecurity information, improving overall detection capability. Additionally, multilingual URL analysis and webpage content analysis could be incorporated to detect sophisticated phishing attacks targeting users in different regions Overall these improvements can transform the proposed system into a comprehensive cybersecurity solution suitable for both desktop and mobile environments.

VII. CONCLUTION

The rapid increase in malicious webpages has created significant cybersecurity challenges for internet users. Traditional detection methods, including blacklist-based and signature-based approaches, are often unable to identify newly emerging malicious websites. This research presented a Real-Time Malicious Webpage Detection System using a Hybrid CNN–LSTM model, which automatically learns structural and sequential characteristics of URLs for accurate classification.The proposed methodology combines the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. CNN efficiently extracts lexical and structural features from URLs, while LSTM captures sequential dependencies that improve classification performance. This combination enables the system to detect malicious webpages more effectively than conventional approaches.The trained model was successfully deployed through a Flask-based web application, providing users with a simple interface for real-time URL verification. In addition, the implementation of a Google Chrome Extension allows users to analyze webpage safety directly during web browsing, improving usability and practical deployment.Experimental evaluation demonstrated that the Hybrid CNN–LSTM model achieves reliable performance in distinguishing safe and malicious webpages. The proposed system contributes to enhancing cybersecurity by providing fast, accurate, and intelligent malicious webpage detection. With future enhancements such as mobile application support, multi-browser compatibility, cloud deployment, and continuous model updates, the system can become a robust solution for protecting users against evolving cyber threats.

REFERENCES

[1] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, K. Rieck, and C. Siemens, "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket," Proc. NDSS, 2014. DOI: 10.14722/ndss.2014.23247.
[2] L. Mariconti, I. Onwuzurike, P. Andriotis, E. De Cristofaro, G. Ross, and G. Stringhini, "MaMaDroid: Detecting Android Malware by Building Markov Chains of Behavioral Models," ACM Transactions on Privacy and Security, vol. 22, no. 2, 2019. DOI: 10.1145/3301295.
[3] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J. Niyigena, "An Effective Phishing Detection Model Based on Character-Level CNN, LSTM and CNN–LSTM," Electronics, vol. 9, no. 12, 2020. DOI: 10.3390/electronics9122012.
[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019. DOI: 10.1109/ACCESS.2019.2895334.
[5] M. Sahingoz, B. Buber, O. Demir, and B. Diri, "Machine Learning Based Phishing Detection from URLs," Expert Systems with Applications, vol. 117, pp. 345–357, 2019. DOI: 10.1016/j.eswa.2018.09.029.
[6] H. Zhu, Z. You, Z. Zhu, W. Shi, X. Chen, and L. Cheng, "DroidDet: Effective and Robust Detection of Android Malware Using Static Analysis Along with Rotation Forest Model," Neurocomputing, vol. 272, pp. 638–646, 2018. DOI: 10.1016/j.neucom.2017.07.030.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, July 2026
Impact Factor: 8.2

[7] W. Wang, Z. Gao, M. Zhao, Y. Li, J. Liu, and X. Zhang, "DroidEnsemble: Detecting Android Malicious Applications with Ensemble of String and Structural Static Features," IEEE Access, vol. 6, pp. 31798–31807, 2018. DOI: 10.1109/ACCESS.2018.2835654.
[8] W. Wang, Y. Li, X. Wang, J. Liu, and X. Zhang, "Detecting Android Malicious Apps and Categorizing Benign Apps with Ensemble of Classifiers," Future Generation Computer Systems, vol. 78, pp. 987–994, 2018. DOI: 10.1016/j.future.2017.01.019.
[9] W. Wang, M. Zhao, and J. Wang, "Effective Android Malware Detection with a Hybrid Model Based on Deep Autoencoder and Convolutional Neural Network," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 8, pp. 3035–3043, 2019. DOI: 10.1007/s12652-018-0803-6.
[10] S. I. Imtiaz, S. U. Rehman, A. R. Javed, Z. Jalil, X. Liu, and W. S. Alnumay, "DeepAMD: Detection and Identification of Android Malware Using High-Efficient Deep Artificial Neural Network," Future Generation Computer Systems, vol. 115, pp. 844–856, 2021. DOI: 10.1016/j.future.2020.10.008.
[11] H. Zhang, S. Luo, Y. Zhang, and L. Pan, "An Efficient Android Malware Detection System Based on Method-Level Behavioral Semantic Analysis," IEEE Access, vol. 7, pp. 69246–69256, 2019. DOI: 10.1109/ACCESS.2019.2919796.
[12] H. Zhu, Y. Li, R. Li, J. Li, Z. You, and H. Song, "SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection," IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 984–994, 2021. DOI: 10.1109/TNSE.2020.2996379.
[13] H. Zhu, L. Wang, S. Zhong, Y. Li, and V. S. Sheng, "A Hybrid Deep Network Framework for Android Malware Detection," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 12, pp. 5558–5570, 2022. DOI: 10.1109/TKDE.2021.3067658.
[14] H. Zhu, W. Gu, L. Wang, Z. Xu, and V. S. Sheng, "Android Malware Detection Based on Multi-Head Squeeze-and-Excitation Residual Network," Expert Systems with Applications, vol. 212, 118705, 2023. DOI: 10.1016/j.eswa.2022.118705.
[15] B. Urooj, M. A. Shah, C. Maple, M. K. Abbasi, and S. Riasat, "Malware Detection: A Framework for Reverse Engineered Android Applications Through Machine Learning Algorithms," IEEE Access, vol. 10, pp. 89031–89050, 2022. DOI: 10.1109/ACCESS.2022.3149053.
[16] M. M. Alani and A. I. Awad, "PAIRED: An Explainable Lightweight Android Malware Detection System," IEEE Access, vol. 10, pp. 73214–73228, 2022. DOI: 10.1109/ACCESS.2022.3189645.
[17] W. Zhang, H. Wang, H. He, and P. Liu, "DAMBA: Detecting Android Malware by ORGB Analysis," IEEE Transactions on Reliability, vol. 69, no. 1, pp. 55–69, 2020. DOI: 10.1109/TR.2019.2924677.
[18] H. Wang, W. Zhang, and H. He, "You Are What the Permissions Told Me! Android Malware Detection Based on Hybrid Tactics," Journal of Information Security and Applications, vol. 66, 103159, 2022. DOI: 10.1016/j.jisa.2022.103159.
[19] L. Wei, W. Luo, J. Weng, Y. Zhong, X. Zhang, and Z. Yan, "Machine Learning-Based Malicious Application Detection of Android," IEEE Access, vol. 5, pp. 25591–25601, 2017. DOI: 10.1109/ACCESS.2017.2771470.
[20] Y. Chen, H. Chen, T. Takahashi, B. Sun, and T.-N. Lin, "Impact of Code Deobfuscation and Feature Interaction in Android Malware Detection," IEEE Access, vol. 9, pp. 123208–123219, 2021. DOI: 10.1109/ACCESS.2021.3110408.