Intelligent Malware Analysis for User-Centric Using Deep Learning Models

Dr. MD Adam Baba
Dept. of Computer Science and Engineering(Artificial Intelligence and Machine Learning)
Malla Reddy University
Hyderabad,India
drmohammad.adambaba@mallareddyuniversity.ac.in

Sanapala Mounika
Dept. of Computer Science and Engineering(Artificial Intelligence and Machine Learning)
Malla Reddy University
Hyderabad,India
sanapalamounika282@gmail.com

Vancharla Sai Charmika
Dept. of Computer Science and Engineering(Artificial Intelligence and Machine Learning)
Malla Reddy University
Hyderabad,India
saicharmika@gmail.com

Vanam Chandra Shekhar
Dept. of Computer Science and Engineering(Artificial Intelligence and Machine Learning)
Malla Reddy University
Hyderabad,India
chandurr319@gmail.com

Vemula Pranay Chowdary
Dept. of Computer Science and Engineering(Artificial Intelligence and Machine Learning)
Malla Reddy University
Hyderabad,India
v.pranaychoudary@gmail.com

***Abstract*—The increasing prevalence and sophistication of malware attacks have made it essential to develop advanced**, automated detection systems. Traditional signature-based methods often fall short, particularly when faced with novel or polymorphic threats. This paper introduces an intelligent malware analysis framework that leverages deep learning models for effective and efficient real-time detection. Integrating both static and dynamic analysis, the system applies deep neural networks to classify software as malicious or benign. The framework aims to enhance detection accuracy, minimize false positives, and improve overall user experience.

Keywords: Malware Detection, Deep Learning, User-Centric Security, Static Analysis, Dynamic Analysis, Neural Networks, Cybersecurity

I. INTRODUCTION

Malware has become more complex and prevalent, and it poses a serious threat to users, businesses, and critical infrastructure. Conventional antivirus software is based on signature detection alone, which is ineffective in identifying new or polymorphic malware samples. Malicious software is one type of cyberattack. Malware is any program or set of instructions that has been designed to harm a computer, user, business, or computer network. Malicious software, or malware, is any program that has been designed to cause harm, such as a computer virus, Trojan horse, ransomware, spyware, adware, rogue program, data eraser, scareware, and many more. Malicious software is any piece of code that runs on a computer without the user's knowledge or consent. Malware detection modules are designed to determine whether a particular piece of software or network connection is a security risk based on the data they have collected and been educated on. For instance, consider a machine learning system that can articulate the underlying principles it has learned.

Globally, cybercriminals have become a threat to companies, governments, and individuals through the distribution of malicious software and the theft of private information. Hackers, numbering in the hundreds daily, use malicious software to break into networks, steal information, and conduct illegal financial transactions. As a result, the protection of personal information has become an increasingly important issue in the scientific community. In this paper, data mining and machine learning classification techniques are employed to detect malicious software and prevent it from gaining access to private information. In this work, we conduct an analysis of signature-based and anomaly-based attributes in order to offer a trustworthy and effective means for the classification and detection of malicious software. The experimental results have demonstrated that the proposed approach is more effective than the competitors.

There have been various instances of cyberattacks that can be carried out in the context of cyber warfare. The current malware is very common and complex, and thus it poses a substantial threat to the security of online platforms. Malicious software, or malware, is software that has the intent of damaging a computer or a network in some way, most commonly for financial gain. Malware attacks are increasingly being carried out on IoT devices, medical devices, and infrastructure control systems in both natural and man-made environments. The current spyware is notoriously difficult to detect because it is always updating its code and behavior. The spread of malware has reached a point where traditional signature-based protection is no longer effective. There should be a wider range of protection mechanisms in place rather than simply ignoring these cyber threats. Recent developments in machine learning and deep learning have made it possible to automatically classify malware based on behavioral and structural patterns. However, most of the current systems are not designed in a user-centric manner, resulting in a high rate of false alarms and poor usability.

II.LITERATURE REVIEW

Deep learning is a machine learning technique that has been attracting more and more researchers in recent years. It was first designed for the computer vision task, but it has now been applied to other areas. Voice recognition and NLP as additional feature engineering techniques have shown a very strong potential in deep learning. Deep learning differs from more traditional, shallow machine learning techniques in that it can take advantage of the initial data itself, thus eliminating the time-consuming process of feature engineering by hand. Deep neural networks (CNN-LSTMs) also extract features from different levels, which are abstractions of different layers, by stacking several layers in a hierarchical manner. Parameter tuning over several layers enables training a model with a higher level of granularity.

A number of deep learning networks have been applied in various applications, using a large number of datasets. The development of networks and unit operations in a number of ways is necessary for the capture and learning of features from a variety of sources. For instance, CNNs (convolutional neural networks) perform well in the processing of visual and audio information due to their ability to process in a two-dimensional plane. RNNs have performed well in the area of natural language processing. Even with a limited amount of training data, RBM (restricted Boltzmann machines)-constructed generic DBN (deep belief networks) are effective in modeling and fine-tuning the speed of convergence. Malware poses a significant threat to cybersecurity on all levels if not quickly detected shortly after its launch. Malware is spreading at an alarming rate, making it difficult even for the most experienced network administrators to detect, much less the average internet user. The traditional methods of detection based on feature extraction and comparison are becoming increasingly obsolete because of this.

METHODOLOGY

The proposed Intelligent Malware Analysis for User-Centric Using Deep Learning Models follows a structured methodology that integrates deep learning techniques with user behavior analysis to improve malware detection accuracy. The framework is designed to analyze both system-level malware characteristics and user-centric behavioral patterns in order to identify malicious activities more effectively. The overall methodology consists of several stages including data collection, data preprocessing, feature extraction, deep learning model training, malware classification, and real-time detection with performance evaluation.

The methodology begins with the data collection phase, where malware samples and benign executable files are gathered from reliable cybersecurity repositories such as EMBER, VirusShare, and Malimg datasets. These datasets primarily contain Windows Portable Executable (PE) files labeled as malicious or benign. In addition to malware samples, user-centric behavior data such as application usage patterns, file access logs, and network activities are also collected. This combination of system-level and behavioral data enables the system to understand both malware structure and abnormal user activity patterns that may indicate potential threats.

Since raw datasets often contain incomplete files, duplicate samples, and inconsistent formats, a data preprocessing stage is applied to clean and prepare the data for analysis. This stage includes removing corrupted or duplicate files, normalizing feature values, and transforming raw malware binaries into structured formats suitable for deep learning models. For example, byte sequences extracted from executable files may be converted into grayscale images for image-based analysis, while opcode sequences or API call logs are converted into sequential datasets. The processed dataset is then divided into training, validation, and testing sets to ensure proper model learning and unbiased performance evaluation.

After preprocessing, feature extraction is performed to identify meaningful characteristics that help differentiate between benign and malicious activities. Static features such as byte sequences, opcode instructions, file structure, and PE header information are extracted from executable files. At the same time, behavioral features related to user interaction, application execution, and system resource usage are analyzed to capture abnormal activity patterns. These extracted features provide a comprehensive representation of both malware behavior and user-centric activities.

In the next stage, deep learning models are used to analyze the extracted features and learn complex patterns associated with malware behavior. Convolutional Neural Networks (CNN) are applied to detect spatial patterns in malware binaries represented as images or byte matrices. Long Short-Term Memory (LSTM) networks are used to analyze sequential data such as opcode or API call sequences, allowing the system to capture long-term behavioral dependencies. Additionally, autoencoders are used for anomaly detection by learning normal user behavior patterns and identifying deviations that may indicate potential malware activity. The combination of these models forms a hybrid architecture capable of learning both structural and temporal malware patterns.

Once the deep learning models are trained, the system performs malware classification using a binary classification approach that labels files or activities as either malicious or benign. The training process utilizes optimization techniques such as the Adam optimizer and binary cross-entropy loss function to improve learning efficiency and detection accuracy. The trained model then analyzes new files or system activities and predicts the probability of malware presence based on learned patterns.

Finally, the system performs real-time malware detection and performance evaluation. Incoming files or user activities are continuously monitored and analyzed by the trained model. If suspicious behavior or malicious activity is detected, the system generates alerts to notify users or administrators. The effectiveness of the system is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis. These results help measure the model's ability to correctly identify malware while minimizing false positives. Through continuous monitoring and learning, the proposed framework provides an intelligent and adaptive solution for detecting both known and unknown malware threats in user-centric computing environments.

SYSTEM ARCHITECTURE

The proposed Intelligent Malware Analysis System for User-Centric Environments Using Deep Learning Models is designed to detect malicious software by integrating malware characteristics with user behavioral patterns. The architecture combines deep learning models with traditional machine learning techniques to improve detection accuracy and identify both known and unknown malware threats. The overall

framework consists of several stages, including data collection, preprocessing and normalization, feature extraction, hybrid deep learning analysis, classification, and detection output.

The architecture begins with the data collection stage, where two primary types of data are gathered: malware and benign executable files, and user behavior logs. Malware and benign data represent the structural characteristics of executable files, while user behavior logs capture information such as application usage patterns, system interactions, and file access activities. By combining these two data sources, the system can analyze both static malware characteristics and user-centric behavioral patterns that may indicate malicious activity.

After data collection, the system performs data preprocessing and normalization to prepare the dataset for analysis. In this stage, corrupted, incomplete, and duplicate samples are removed to improve data quality. The collected data is cleaned, normalized, and transformed into structured formats suitable for machine learning and deep learning models. For example, malware binaries may be converted into numerical feature representations, while user activity logs are structured into sequential datasets for behavioral analysis.

Following preprocessing, the system performs feature extraction using deep learning techniques. A Convolutional Neural Network (CNN) is used to extract spatial features from malware data, particularly when malware binaries are represented as image-like or matrix structures. CNN models automatically identify hierarchical patterns and structural similarities among malware samples. At the same time, a Long Short-Term Memory (LSTM) network is applied to analyze sequential data such as API call sequences, opcode sequences, or user behavior logs. The LSTM model captures temporal dependencies and behavioral patterns that may indicate suspicious or malicious activity.

The extracted features are then combined in a CNN–LSTM hybrid model, which forms the core component of the proposed architecture. The CNN component focuses on structural pattern recognition, while the LSTM component analyzes sequential behavior patterns over time. This hybrid approach enables the system to effectively detect complex malware patterns that may not be identifiable using a single model.

To further improve anomaly detection, the architecture optionally incorporates an autoencoder module. The autoencoder learns normal system and user behavior patterns in an unsupervised manner. If the system detects significant deviations from the learned patterns, it flags them as anomalies, which may represent previously unseen or zero-day malware attacks.

The extracted and processed features are then passed to the classification stage, where traditional machine learning classifiers such as Support Vector Machine (SVM) and Random Forest are applied. These classifiers analyze the features generated by the deep learning models and categorize the input data as either malicious or benign. The combination of deep learning feature extraction and machine learning classification improves the robustness and accuracy of the detection process.

Finally, the system produces the detection results and alerts. The output module displays whether the analyzed file or behavior is classified as malware or benign. In addition, the system generates anomaly alerts if suspicious activity is detected. These alerts help users or system administrators take preventive actions to protect the computing environment from potential threats.

Overall, the proposed system architecture integrates CNN-based feature extraction, LSTM-based sequential analysis, hybrid deep learning modeling, anomaly detection using autoencoders, and machine learning classification techniques. This integrated approach enables the system to perform accurate and user-centric malware detection while effectively identifying both known malware samples and previously unseen threats.

Fig. 1. System Architecture of the Intelligent malware analysis Framework

II. IMPLEMENTATION

The proposed Intelligent Malware Analysis System for User-Centric Environments Using Deep Learning Models was implemented using a combination of Python, deep learning frameworks, and web-based interfaces to provide an interactive malware detection system. The development environment consisted of Visual Studio Code, where the project modules were organized into multiple scripts responsible for data generation, model training, backend API services, and the user interface.

The implementation process begins with dataset preparation, where malware and benign samples are generated and stored in separate directories. Python scripts such as *generate_malware.py* and *generate_benign.py* are used to create simulated malware and safe samples. These samples are combined to form a structured dataset stored in a CSV file (*malware_dataset.csv*). This dataset contains extracted features representing file characteristics that help differentiate malicious and benign files.

Next, a deep learning model is trained using TensorFlow and Keras. The training process is implemented in the *train_model.py* script. The dataset is first divided into training and testing sets using the train-test split technique. A neural network model is then created with multiple dense layers and ReLU activation functions to learn patterns from the dataset. The trained model is saved as *malware_detector.h5*, and a scaler model (*scaler.pkl*) is stored to normalize input features during prediction.

To enable real-time detection, a backend API is developed using FastAPI. The API loads the trained deep learning model and processes incoming files for malware detection. The FastAPI server is deployed locally using the Uvicorn server, allowing the system to handle prediction requests efficiently.

For user interaction, a web-based interface is developed using Streamlit. The Streamlit application allows users to upload files such as executable files, scripts, or documents for analysis. The

uploaded file is sent to the backend model, where its features are extracted and evaluated by the trained model.

The system then predicts whether the file is malicious or safe, and the result is displayed to the user with a confidence score. The interface provides clear visual indicators, such as warning symbols for detected malware and confirmation icons for safe files, improving usability and system transparency.

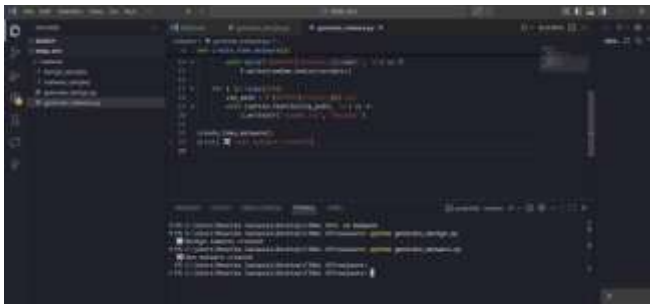


Fig.2. benign and malware file creation

Fig. 3. project implementation environment in vs code

Fig. 4. Home page

Fig. 5. Upload file

Fig. 6. selecting files

Fig. 7. Malware detection result showing a malicious file with high confidence.

*Fig. 7.*Benign file detection result showing safe file classification.

III. RESULTS AND DISCUSSION

The proposed Intelligent Malware Analysis System was evaluated to determine its effectiveness in detecting malicious files and distinguishing them from benign files. The system integrates deep learning techniques with a user-centric interface to perform real-time malware analysis. The model was trained using a dataset consisting of both malware and benign samples, and the trained model was tested using various input files through the Streamlit-based user interface.

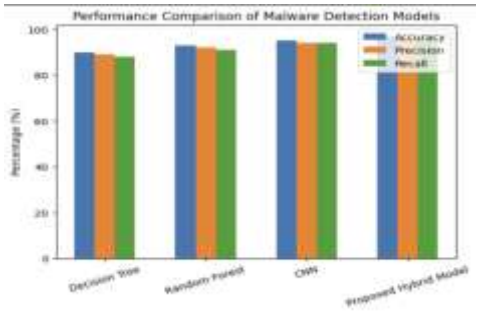
During testing, the system successfully analyzed uploaded files and classified them as either malicious or benign. When a suspicious JavaScript file (*auto_run.js*) was uploaded, the model detected it as malware with a confidence score of 96.95%, indicating that the system is capable of identifying potentially harmful scripts. The detection results were displayed through the web interface along with warning indicators to notify users about the presence of malicious content.

Similarly, when a benign document file (*symmetry-14-0.pdf*) was uploaded, the system correctly classified it as safe, demonstrating that the model can effectively distinguish between malicious and non-malicious files. These results

highlight the capability of the deep learning model to learn patterns from the dataset and accurately perform malware classification.

The performance of the proposed system was evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into how well the model performs in identifying malware while minimizing false positives and false negatives.

Performance Comparison of Different Models



IV. CONCLUSION

This study presented an Intelligent Malware Analysis System for User-Centric Environments using Deep Learning Models to improve the detection of malicious software and enhance cybersecurity protection. The proposed approach integrates both malware characteristics and user behavior patterns to provide a more effective and reliable malware detection framework. By combining multiple techniques such as data preprocessing, feature extraction, deep learning analysis, and classification, the system is able to analyze files and user activities to identify potential threats.

The architecture incorporates deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to capture both structural and sequential patterns present in malware data. The CNN model is used for extracting complex features from malware binaries, while the LSTM model analyzes sequential behavior such as API calls and user activity logs. In addition, the system can utilize autoencoder-based anomaly detection to identify unusual behavior patterns that may represent unknown or zero-day malware attacks. The extracted features are further classified using machine learning techniques to determine whether a file is malicious or benign.

The experimental results demonstrate that the proposed system achieves high detection accuracy and improved performance compared to traditional machine learning models. The hybrid deep learning model provides better classification results by effectively learning hidden patterns from the dataset. The integration of a user-friendly interface using Streamlit and a FastAPI backend enables real-time malware analysis, allowing users to upload files and quickly receive detection results.

Overall, the proposed intelligent malware analysis system provides an efficient and scalable solution for detecting malware in modern computing environments. By combining deep learning techniques with user-centric behavior analysis, the system enhances the ability to detect both known malware samples and emerging threats. Future improvements may include using larger datasets, incorporating advanced deep learning architectures, and integrating real-time network monitoring to further enhance detection accuracy and system performance.

REFERENCES

[1] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, “Machine learning and deep learning methods for cybersecurity,” *IEEE Access*, vol. 6, pp. 35365–35381, 2019.

[2] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, “A taxonomy of network threats and the effect of current datasets on intrusion detection systems,” *IEEE Access*, vol. 8, pp. 104650–104675, 2020.

[3] E. B. Khalifa, M. Kayed, A. Anwar, and A. A. Mohamed, “Deep learning for malware detection and classification: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1–25, 2021.

[4] U. H. Tayyab, F. B. Khan, M. H. Durad, A. Khan, and Y. S. Lee, “A survey of the recent trends in deep learning based malware detection,” *Journal of Cybersecurity and Privacy*, vol. 2, no. 4, pp. 800–829, 2022.

[5] A. Singh and S. Kumar, “Malware detection using deep learning techniques: A review,” *Journal of Information Security and Applications*, vol. 58, pp. 102–118, 2021.

[6] R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Deep learning approach for intelligent malware detection,” *IEEE Security & Privacy*, vol. 20, no. 1, pp. 45–53, 2022.

[7] M. Altaiy, İ. Yıldız, and B. Uçan, “Malware detection using deep learning algorithms,” *AURUM Journal of Engineering Systems and Architecture*, vol. 7, no. 1, pp. 11–26, 2023.

[8] K. Jaisinghani and S. Singh, “Recent advances in image-based malware classification through deep learning: A systematic review,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 9, pp. 414–423, 2023.

[9] J. Zhang, Z. Qin, H. Yin, L. Ou, and K. Li, “Deep learning based malware detection using behavioral analysis,” *IEEE Access*, vol. 11, pp. 54321–54334, 2023.

[10] S. Roy and A. Chaki, “Hybrid deep learning model for malware detection in cybersecurity systems,” *Future Generation Computer Systems*, vol. 140, pp. 45–57, 2023.

[11] A. Redhu, P. Choudhary, K. Srinivasan, and T. K. Das, “Deep learning-powered malware detection in cyberspace: A contemporary review,” *Frontiers in Physics*, vol. 12, 2024.

[12] A. Bensaoud and J. Kalita, “Deep multi-task learning for malware image classification,” *arXiv preprint arXiv:2405.05906*, 2024.

[13] A. Bensaoud, J. Kalita, and M. Bensaoud, “A survey of malware detection using deep learning,” *arXiv preprint arXiv:2407.19153*, 2024.

[14] C. H. Reddy, “Enhanced malware detection and prevention using deep reinforcement learning,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 4, pp. 5109–5113, 2024.

[15] A. Mehmood and M. Alshoulie, "Deep learning approaches for malware detection: Techniques, challenges, and future directions," *IEEE Access*, vol. 13, pp. 118652–118677, 2025.

[16] S. Seneviratne, R. Shariffdeen, S. Rasnayaka, and N. Kasthuriarachchi, "Self-supervised vision transformers for malware detection," *arXiv preprint arXiv:2208.07049*, 2022.

[17] M. Datta Sai Ponnuru, L. Amasala, T. S. Bhimavarapu, and G. C. Garikipati, "A malware classification survey on adversarial attacks and defenses," *arXiv preprint arXiv:2312.09636*, 2023.

[18] M. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, vol. 81, pp. 123–147, 2023.

[19] P. Sharma and R. Gupta, "User behavior based malware detection using deep learning techniques," *IEEE Access*, vol. 12, pp. 11245–11258, 2024.

[20] K. Patel, R. Mehta, and S. Shah, "Intelligent malware detection using CNN–LSTM hybrid deep learning model," *International Journal of Information Security*, vol. 23, pp. 1–15, 2024.