IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511142

Machine Learning Based Framework for Ransomware Detection and Classification

Gnanamani H[1], Akshata G B[2], Ananya K[3], Anju K[4], Anusha[5]

Assistant Professor, Department of CSE, East West College of Engineering, Bengaluru, India[1]

Student, Department of CSE, East West College of Engineering, Bengaluru, India[2]

Student, Department of CSE, East West College of Engineering, Bengaluru, India[3]

Student, Department of CSE, East West College of Engineering, Bengaluru, India[4]

Student, Department of CSE, East West College of Engineering, Bengaluru, India[5]

Abstract: Ransomware is a type of malicious software that encrypts important user files and demands a ransom to restore access, posing a serious threat to individuals, businesses, and organizations. Traditional antivirus systems primarily rely on signature-based detection techniques, which may fail to identify new and unknown ransomware variants. To address this challenge, this paper proposes a Machine Learning-Based Framework for Ransomware Detection and Classification that uses behavior-based analysis to identify potentially malicious files. The proposed framework involves dataset collection, data preprocessing, feature extraction, and machine learning model training. Relevant features, including file size, file entropy, API calls, file operations, CPU usage, memory usage, and network activity, are analyzed to identify patterns associated with ransomware behavior. A Random Forest classifier is used to classify input files or their extracted features as benign or potentially malicious. The trained model is integrated with a Flask-based web application that allows users to submit files for analysis and view the detection results. The framework also categorizes detected threats into high, medium, and low risk levels to help users understand potential security risks. The proposed approach aims to improve the identification of suspicious files, support early threat detection, and reduce the risk of data loss. This framework demonstrates the potential of machine learning in strengthening cybersecurity and provides a foundation for future enhancements, including real-time monitoring, improved classification models, and automated threat response mechanisms.

Keywords: Ransomware Detection, Machine Learning, Random Forest, Feature Extraction, File Entropy, Cybersecurity, Flask, Threat Classification.

I.INTRODUCTION

Ransomware is a type of malicious software that encrypts important files and data on a computer system and demands a ransom from victims to restore access. Ransomware attacks have become a major cybersecurity threat to individuals, businesses, and organizations. These attacks can cause data loss, financial damage, and disruption of essential services. As cyber threats continue to evolve, detecting ransomware accurately and at an early stage has become increasingly important.

Traditional antivirus systems mainly rely on signature-based detection, which identifies malware by comparing files against previously known patterns. However, these methods may fail to detect new or modified ransomware variants whose signatures are not available in existing databases. Therefore, more adaptive and intelligent detection techniques are required to identify unknown threats.

Machine learning (ML) provides a promising approach to ransomware detection by analyzing file characteristics and system behavior to identify suspicious patterns. Features such as file size, file entropy, API calls, file operations, CPU usage, memory usage, and network activity can help distinguish potentially malicious activities from normal operations. Machine learning algorithms can learn from datasets containing benign and ransomware-related samples and use the learned patterns to classify new inputs.

This paper presents a Machine Learning-Based Framework for Ransomware Detection and Classification using a Random Forest algorithm. The proposed system includes dataset collection, data preprocessing, feature extraction, model training, and classification. A Flask-based web application provides a user interface through which users can submit files for analysis and view the detection results. The system also categorizes potential threats into high, medium, and low risk levels to make the results easier to understand.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511142

The main aim of this framework is to support the identification of suspicious files, improve ransomware detection capabilities, and reduce the potential risk of data loss. The proposed approach demonstrates how machine learning can contribute to cybersecurity and provides a foundation for future improvements, including real-time monitoring, enhanced classification techniques, and automated threat response.

II.RELATED WORK

Ransomware has become one of the major cybersecurity threats because it encrypts important files and demands payment from victims to restore access. Traditional antivirus systems mainly depend on signature-based detection techniques to identify known ransomware threats. However, these methods may fail to detect new, modified, or previously unknown ransomware attacks.
Researchers have explored different approaches to improve ransomware detection using machine learning, behavioural analysis, and data mining techniques. These approaches analyse file characteristics and system activities to identify suspicious patterns.

A. Traditional Signature-Based Detection

Traditional ransomware detection systems use predefined signatures to identify malicious files. These signatures are compared with incoming files to determine whether they contain known ransomware patterns. Although this approach is useful for detecting previously identified threats, it has limitations when dealing with new ransomware variants. Modified ransomware may change its signature and avoid detection.

B. Behaviour-Based Detection

Behaviour-based detection identifies ransomware by analysing its activities during execution. It monitors activities such as file modifications, API calls, file access operations, CPU and memory usage, and network connections. Unusual behaviour, such as rapid file modifications or suspicious encryption activities, may indicate a ransomware attack. However, behavioural detection may require continuous monitoring and sufficient system resources.

C. Machine Learning-Based Detection

Machine learning techniques provide an effective approach to identifying ransomware by learning patterns from previously collected data. These techniques analyse different features of files and system activities to classify them as benign or malicious. Algorithms such as Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, and Neural Networks can be used for ransomware detection.
Machine learning-based detection can identify patterns that may not be detected by traditional signature-based methods. However, its performance depends on the quality, quantity, and diversity of the training dataset.

D. Random Forest-Based Classification

Random Forest is a supervised machine learning algorithm that combines multiple decision trees to produce a final prediction. It analyses input features and determines the appropriate class based on the predictions of individual trees.
Random Forest is useful for ransomware classification because it can analyse multiple features and identify relationships between them. In the proposed project, the Random Forest algorithm is used to classify input records into benign and ransomware categories.

E. Research Gap

Existing ransomware detection approaches may face challenges in identifying new ransomware variants and analysing different suspicious activities. Therefore, the proposed project focuses on developing a machine learning-based framework that analyses selected features and classifies suspicious activity into benign or ransomware categories.

III.PROPOSED SYSTEM

The proposed system, Machine Learning Based Framework for Ransomware Detection and Classification, aims to identify and classify ransomware using machine learning techniques. The system analyses important features associated with files and system activities to determine whether the input is benign or ransomware-related.

The proposed framework uses Python for implementation and the Random Forest algorithm for classification. Data preprocessing and feature extraction are performed before training the machine learning model. The trained model analyses the input features and generates a prediction. The system also provides a Flask-based web interface through which users can interact with the detection framework and view the classification results.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

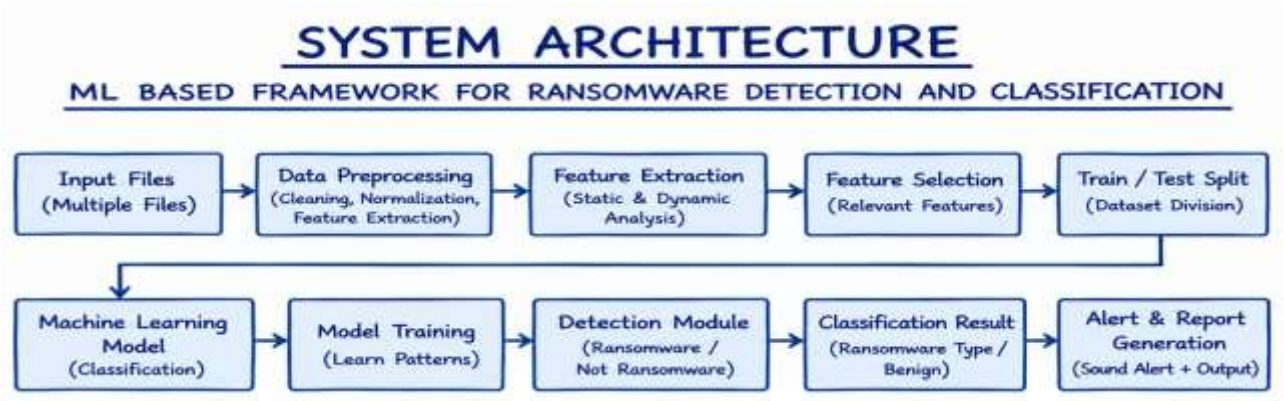
DOI: 10.17148/IJARCCE.2026.1511142

A. System Architecture

The proposed system consists of the following major modules:

- Dataset Collection Module: Collects labelled data containing benign and ransomware-related records for model training and testing.
- Data Preprocessing Module: Cleans and prepares the collected data by handling missing values, removing unnecessary information, and preparing features for machine learning.
- Feature Extraction Module: Extracts important features such as file size, entropy, API calls, system resource usage, and network connections.
- Machine Learning Module: Uses the Random Forest algorithm to learn patterns from the prepared training data. Detection and Classification Module: Analyses the input features and predicts whether the input belongs to the benign or ransomware category.
- Result Display Module: Displays the prediction and detection information through the Flask-based web application.

The overall system provides a structured approach to detecting ransomware using machine learning and presenting the results in an understandable format**.**



IV. IMPLEMENTATION AND METHODOLOGY

A. System Implementation

The proposed ransomware detection framework is implemented using Python and machine learning techniques.
Python libraries such as Pandas, NumPy, and Scikit-learn are used for data processing and model development. Flask is used to provide a web-based interface for user interaction.
The Random Forest algorithm is used as the primary classification algorithm. The implementation follows a modular approach consisting of dataset collection, data preprocessing, feature extraction, model training, testing, prediction, and result display.
The main objective of implementation is to identify suspicious ransomware-related patterns and classify the input into benign or ransomware categories**.**

B. Dataset Collection

Dataset collection is the initial stage of the proposed system. It involves collecting labelled records representing normal and ransomware-related activities.
The collected data is used to train and evaluate the machine learning model. The dataset contains different features that help the model distinguish between normal and suspicious behaviour.
The quality of the dataset plays an important role in the accuracy and reliability of ransomware detection.

C. Data Preprocessing

Data preprocessing is performed to improve the quality and consistency of the collected dataset.
The following operations are performed:

- Handling missing or incomplete values.
- Removing unnecessary and duplicate records.
- Preparing the required input features.
- Converting categorical information into a suitable format.
- Separating input features and target labels.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511142

- Dividing the dataset into training and testing sets.
- These operations help prepare the data for machine learning model development.

D. Feature Extraction

Feature extraction is an important stage in ransomware detection. It identifies relevant characteristics that help distinguish normal activities from ransomware-related activities.
The framework considers features such as:

- File Size: Represents the size of the file being analysed.
- Entropy: Measures the randomness of data. High entropy may indicate encrypted or compressed content.
- API Calls: Represents application programming interface operations that may provide information about program behaviour.
- CPU Usage: Indicates the processing resources used by an application.
- Memory Usage: Represents the amount of memory consumed during execution.
- Network Connections: Provides information about communication activities performed by a program.
- These features are used by the classification model to identify suspicious patterns.

E. Random Forest Classification

Random Forest is the main machine learning algorithm used in the proposed system.
It works by creating multiple decision trees using the training dataset. Each decision tree produces a prediction, and the final classification is determined through the majority voting process.

The model learns the relationship between the extracted features and their corresponding labels. During prediction, it analyses the input data and classifies it into one of the following categories:

- Benign: Indicates normal or non-malicious activity.
- Ransomware: Indicates ransomware-related or malicious activity.

The trained model is used to predict the category of new input records.

F. Model Training and Testing

The collected dataset is divided into training and testing data. The training data is used to teach the Random Forest model to identify different patterns.
The testing data is then provided to evaluate the model's classification performance. Evaluation measures such as accuracy, precision, recall, F1-score, and confusion matrix can be used to analyse the performance of the trained model.

G. Web Application Development

A Flask-based web application is developed to provide an interface for interacting with the ransomware detection framework. The application connects the trained machine learning model with the user interface. The prediction generated by the model is displayed to the user as a detection result.
This makes the system easier to access and understand without requiring users to execute machine learning commands manually.

METHODOLOGY

The methodology of the proposed system follows a sequence of operations that begins with dataset collection and ends with ransomware detection and classification.

Step 1: Dataset Collection
Collect labelled data containing benign and ransomware-related records.

Step 2: Data Preprocessing
Clean the collected data by handling missing values, removing duplicates, and preparing the required features.

Step 3: Feature Extraction
Extract important features such as file size, entropy, API calls, CPU usage, memory usage, and network connections.

Step 4: Dataset Splitting
Divide the prepared dataset into training and testing sets for model development and evaluation.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511142

Step 5: Model Training
Train the Random Forest classifier using the training dataset to learn patterns associated with benign and ransomware activities.

Step 6: Model Testing
Evaluate the trained model using testing data to measure its classification performance.

Step 7: Ransomware Detection and Classification
Provide input features to the trained model. The model predicts whether the input belongs to the benign or ransomware category.

Step 8: Result Display
Display the classification result and relevant detection information through the Flask web application.

V.RESULTS AND DISCUSSION

The proposed system demonstrates the use of machine learning techniques for ransomware detection and classification. The Random Forest model analyses the extracted features and predicts whether the given input belongs to the benign or ransomware category.

A. Model Performance

The trained Random Forest model is evaluated using classification performance measures. The project output displays an accuracy of 100.0% in the available evaluation result, along with a classification report.
Accuracy represents the percentage of correctly classified records out of the total tested records.
Other evaluation measures include:

- Precision: Measures how many records predicted as ransomware are actually ransomware.
- Recall: Measures how many actual ransomware records are correctly identified.
- F1-Score: Represents the balance between precision and recall.
- Confusion Matrix: Shows the number of correct and incorrect predictions for each class.

The displayed accuracy indicates the performance achieved in the current evaluation. However, further testing with larger and more diverse datasets is required to establish the model's ability to detect previously unseen ransomware.

B. Benign and Ransomware Classification

The system classifies the input into two categories.

- Benign Classification: When the input features match normal activity patterns, the model predicts the input as benign.
- Ransomware Classification: When the input features match ransomware-related patterns, the model predicts the input as ransomware.

This classification helps users understand whether the analysed input is normal or potentially malicious.

C. Web Application Results

The Flask-based application provides an interface for displaying ransomware detection results. The trained model is connected to the application, allowing predictions to be generated and displayed.
The result interface presents the detection outcome in a simple and understandable manner. It helps users identify whether the analysed input is classified as benign or ransomware.

D. Overall Discussion

The proposed framework combines data preprocessing, feature extraction, and Random Forest classification to detect ransomware-related patterns. The system demonstrates how machine learning can be applied to cybersecurity problems. It provides a structured detection process and displays the classification results through a web application.
The overall performance depends on the quality of the dataset, the selected features, and the ability of the trained model to identify different ransomware patterns.

VI.LIMITATIONS

Although the proposed system provides a machine learning-based approach for ransomware detection, certain limitations may affect its performance and reliability.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511142

1. **Dataset Dependency**: The performance of the model depends on the quality and diversity of the training dataset. Limited data may reduce its ability to identify new ransomware variants.
2. **Detection Accuracy**: The model may produce incorrect predictions when it encounters unfamiliar patterns or previously unseen ransomware samples.
3. **New Ransomware Variants**: Newly developed ransomware may use different techniques that are not represented in the training dataset.
4. **Feature Dependency**: The accuracy of classification depends on the quality and availability of extracted features. Missing or incorrect feature values may affect predictions.
5. **Resource Requirements**: Data processing and model training may require sufficient memory and processing resources, especially when handling large datasets.
6. **Limited Real-Time Monitoring**: The current framework focuses on analysing input data and generating predictions. Continuous monitoring of all system activities would require additional implementation.
7. **Model Generalization:** A high accuracy on the available test dataset does not guarantee the same performance on every real-world ransomware attack.

VII.CONCLUSION AND FUTURE SCOPE

A. Conclusion

The project, Machine Learning Based Framework for Ransomware Detection and Classification, presents a machine learning-based approach to identifying ransomware-related activities.
The proposed system uses Python, data preprocessing, feature extraction, and the Random Forest classification algorithm to distinguish between benign and ransomware categories.

Important features such as file size, entropy, API calls, CPU usage, memory usage, and network connections are considered for identifying suspicious patterns. The trained model analyses these features and generates a classification result. The integration of the trained model with a Flask-based web application provides a simple interface for displaying detection results. The project demonstrates the potential of machine learning techniques in improving ransomware detection and supporting cybersecurity applications.

Although the system has certain limitations related to dataset quality, detection accuracy, and unseen ransomware variants, it provides a foundation for developing more advanced ransomware detection frameworks.
Overall, the proposed system aims to improve the identification and classification of ransomware through machine learning techniques and contribute towards better protection against cybersecurity threats**.**

B. Future Scope

The proposed framework can be further enhanced by integrating advanced machine learning techniques and additional security features.
The following improvements can be considered for future development:

- **Larger Dataset**: The system can be trained using larger and more diverse datasets containing different ransomware variants to improve detection reliability.
- **Advanced Machine Learning Algorithms**: Additional algorithms such as Support Vector Machine, Decision Tree, XG Boost, and Neural Networks can be explored and compared to improve classification performance.
- **Real-Time Ransomware Detection:** The framework can be extended to monitor system activities continuously and identify suspicious behaviour during execution.
- **Ransomware Family Classification**: The system can be enhanced to identify and classify different ransomware families instead of only distinguishing between benign and ransomware categories.
- **Improved Feature Extraction:** Additional behavioural features can be included to improve the identification of unknown ransomware patterns.
- **Cloud-Based Integration**: The application can be extended into a cloud-based security platform to provide accessible ransomware detection services.
- **Automated Security Alerts:** The system can generate alerts whenever suspicious ransomware-related activities are detected.
- **Advanced Deep Learning Techniques**: Deep learning models can be explored to identify complex ransomware patterns and improve detection capabilities.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511142

- **Regular Model Updates**: The training dataset and machine learning model can be updated regularly to adapt to newly emerging ransomware threats.

REFERENCES

[1] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and Drop It): Stopping ransomware attacks on user data," in Proceedings of the IEEE 36th International Conference on Distributed Computing Systems (ICDCS), 2016.
[2] A. Kharraz, S. Arshad, C. Mulliner, W. K. Robertson, and E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware," in Proceedings of the 25th USENIX Security Symposium, 2016.
[3] A. Continella et al., "ShieldFS: A self-healing, ransomware-aware filesystem," in Proceedings of the 32nd Annual Conference on Computer Security Applications (ACSAC), 2016.
[4] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," 2016.
[5] F. Andronio, S. Zanero, and L. Maggi, "Heldroid: Dissecting and detecting mobile ransomware," in Proceedings of the 18th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 2015.
[6] D. Maiorca, F. Mercaldo, G. Giacinto, C. A. Visaggio, and F. Martinelli, "R-Pack Droid: API package-based characterization and detection of mobile ransomware," in Proceedings of the ACM Symposium on Applied Computing, 2017.
[7] J. Zhang et al., "Classification of ransomware families with machine learning based on N-gram of opcodes," Future Generation Computer Systems, vol. 90, pp. 211–221, 2019.
[8] K. Lee, S.-Y. Lee, and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," IEEE Access, vol. 7, pp. 110205–110215, 2019.
[9] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in Proceedings of the International Conference on Communication Systems and Networks (COMSNETS), 2018.
[10] A. O. Almashhadani, M. Kaiiali, S. Sezer, and P. O'Kane, "A multi-classifier network-based crypto ransomware detection system: A case study of Locky ransomware," IEEE Access, 2019.
[11] R. Vinayakumar, K. P. Soman, K. K. Senthil Velan, and S. S. Ganorkar, "Evaluating shallow and deep networks for ransomware detection and classification," in Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.
[12] J. Hwang, J. Kim, S. Lee, and K. Kim, "Two-stage ransomware detection using dynamic analysis and machine learning techniques," Wireless Personal Communications, vol. 112, pp. 2597–2609, 2020.
[13] A. K. K. Ramesh and R. Menen, "Ransomware early detection by the analysis of file sharing traffic," 2018.
[14] E. Berrueta, E. Magaña, D. Morató, and M. Izal, "A survey on detection techniques for cryptographic ransomware," IEEE Access, 2019.
[15] A. Alraizza and A. Algarni, "Ransomware detection using machine learning: A survey," Big Data and Cognitive Computing, vol. 7, no. 3, Art. no. 143, 2023.
[16] U. Urooj, B. A. S. Al-rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," Applied Sciences, vol. 12, no. 1, Art. no. 172, 2022.

 This work is licensed under a Creative Commons Attribution 4.0 International License