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. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141104

Ransomware Attacks: Mitigation and Prevention Strategies

Mr. Umesh Manohar Badgujar [1], Prof. Kaustubh Bhave[2], Prof. Manoj V. Nikum*[3]

Student Of MCA, Shri Jaykumar Rawal Institute of Technology Dondaicha, Maharashtra, India[1]

Assistant Professor, Prof MCA Department, Shri Jaykumar Rawal Institute of Technology Dondaicha, Maharashtra, India[2]

Assistant Professor & HOD, MCA Department, Shri Jaykumar Rawal Institute of Technology Dondaicha, Maharashtra, India[3]

Abstract: Ransomware has become one of the most destructive and financially damaging cyber threats, capable of encrypting critical data and demanding ransom payments for recovery. Traditional antivirus solutions based on static signatures are often ineffective against newly emerging or polymorphic ransomware variants. To address these limitations, this research presents a Machine Learning (ML)–based Ransomware Detection and Mitigation Framework implemented using Python.

The proposed system performs behavioral analysis of file operations and process activities, monitoring parameters such as file-write frequency, entropy levels, extension modifications, CPU utilization, and process lineage. These behavioral features are used to train a Random Forest classifier that distinguishes between normal user operations and ransomware-like activity. The trained model, integrated with Python's Watchdog library, continuously monitors the file system in real time and automatically quarantines or isolates suspicious files upon detection.

Additionally, the framework incorporates backup and recovery mechanisms that periodically create immutable file snapshots, ensuring data integrity and supporting post-attack restoration. The combination of Python's data processing ecosystem and ML algorithms provides a scalable, adaptive, and proactive defense mechanism against evolving ransomware threats.

Experimental evaluations demonstrate that the model effectively detects ransomware-like behavior with high accuracy and minimal false positives. Overall, this work contributes to a practical, intelligent, and automated cybersecurity framework for ransomware prevention, early detection, and mitigation, thereby reducing potential data loss and enhancing system resilience against modern cyberattacks.

Keywords: Ransomware, Machine Learning, Cybersecurity, Python, Random Forest, File Behavior Analysis, Data Protection, Threat Mitigation.

I. INTRODUCTION

In the modern digital era, data has become the most valuable asset for individuals, organizations, and governments alike. However, the rapid growth of information technology and internet connectivity has also led to an alarming rise in cybersecurity threats, among which ransomware attacks have emerged as one of the most destructive. Ransomware is a form of malicious software that encrypts users' files or entire systems, rendering them inaccessible until a ransom is paid to the attacker. Such attacks cause significant financial losses, data breaches, and service disruptions across all sectors.

According to recent cybersecurity reports, ransomware incidents have increased by over 80% globally in the last few years, targeting critical infrastructure, healthcare, and educational institutions. Attackers often exploit vulnerabilities, phishing emails, and weak network defenses to infiltrate systems. Modern ransomware variants use advanced encryption techniques and polymorphic code, making them difficult to detect using traditional signature-based antivirus systems.

Conventional defensive measures rely on predefined rules or static patterns, which are ineffective against zero-day ransomware or rapidly evolving attack methods. Therefore, there is a growing need for intelligent, adaptive, and automated detection systems that can identify suspicious behaviors before encryption occurs.

To address these challenges, this project proposes a Machine Learning (ML)–based Ransomware Detection and Mitigation Framework developed using Python. The system performs behavioral analysis of file and process activities, monitoring indicators such **as** file-write rates, entropy levels, extension modifications, and CPU utilization**.** A Random Forest classifier is trained on both synthetic and real-world datasets to distinguish between normal user operations and ransomware-like behavior. Once suspicious activity is detected, the system automatically quarantines files**,** isolates affected processes**,** and triggers backup and recovery mechanisms using immutable snapshots.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141104

By leveraging Python's robust data science ecosystem and the predictive power of machine learning, this framework aims to provide a proactive, transparent, and efficient defense mechanism against ransomware attacks. The proposed approach enhances early threat detection, reduces data loss, and strengthens organizational resilience against evolving cyber threats.

II. LITERATURE REVIEW

Ransomware has emerged as one of the most severe and fast-evolving cyber threats in the digital era. It encrypts user or organizational data and demands payment, often in cryptocurrencies, to restore access. Traditional defense mechanisms such as signature-based antivirus systems and firewalls have proven inadequate against modern ransomware, which continuously mutates and employs advanced evasion techniques. Consequently, researchers have turned to Machine Learning (ML) and Artificial Intelligence (AI) to develop intelligent, adaptive, and automated ransomware detection and prevention systems.

Early studies in ransomware detection primarily relied on static analysis, focusing on identifying malicious code patterns or binary signatures. For instance, Nataraj et al. (2019) used opcode sequences and byte n-grams extracted from binary executables to classify ransomware. Although these methods achieved high accuracy on known variants, they were ineffective against polymorphic or zero-day ransomware due to their reliance on predefined signatures.

To overcome these limitations, researchers began adopting dynamic behavioral analysis**,** which monitors program activity during execution to identify suspicious behavior such as rapid file encryption, registry modification, or unusual process creation. Sgandurra et al. (2020) introduced "EldeRan," a dynamic ransomware detection framework that used Random Forest algorithms on API call sequences and system behaviors, achieving more than 96% accuracy. Similarly, Vinayakumar et al. (2021) applied Long Short-Term Memory (LSTM) networks to analyze time-series behavioral data, effectively identifying zero-day ransomware with minimal false positives.

In addition to these methods, Convolutional Neural Networks (CNNs) have been used to detect ransomware by converting binary files into grayscale images, enabling visual pattern recognition. Chen et al. (2022) developed a CNN-based detection system that outperformed traditional ML classifiers, achieving an accuracy of 98.7% in detecting ransomware families. Deep learning models like CNN and LSTM eliminate the need for manual feature engineering and can automatically learn complex patterns, making them ideal for real-time detection.

Another important research direction involves feature-based ML detection using Python frameworks such as scikit-learn and TensorFlow**.** Islam et al. (2022) used statistical and behavioral features—like file entropy, process activity, and encryption rates—to train Random Forest and Gradient Boosting classifiers, achieving superior detection accuracy with Python-based implementations. Python's flexibility and extensive libraries (Pandas, NumPy, Scikit-learn, TensorFlow, and PyTorch) make it an ideal language for developing scalable ransomware detection systems.

Recent literature also highlights the integration of Explainable AI (XAI) to address the "black-box" nature of deep learning models. Xu and Zhang (2024) utilized SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) techniques to interpret ransomware predictions, allowing security analysts to understand which behavioral features most influenced the model's decision. This transparency enhances trust and helps in identifying false positives or model biases.

Moreover, some studies emphasize real-time mitigation and prevention mechanisms**.** Gupta and Sharma (2024) proposed a Python-based real-time ransomware prevention framework that integrates ML-based detection with automatic process isolation and file backup restoration. The system continuously monitors process behavior, terminates suspected ransomware activities, and restores affected files from backups — providing a proactive defense mechanism.

Hybrid ML frameworks combining multiple algorithms have also gained attention**.** Ali et al. (2024) proposed a hybrid CNN–Random Forest model that merges deep learning feature extraction with ensemble-based decision-making, resulting in higher robustness against diverse ransomware families. Similarly, Singh et al. (2025) introduced an Autoencoder–based anomaly detection approach capable of identifying ransomware by learning normal system behavior patterns and flagging deviations as potential attacks.

From the reviewed literature, it is evident that the evolution of ransomware mitigation strategies has progressed from static signature-based detection to intelligent, adaptive, and explainable ML-powered systems**.** Combining Python's computational capabilities with modern ML and DL techniques enables the development of flexible, scalable, and interpretable frameworks for ransomware detection and prevention. This research builds upon these advancements by proposing a Python-based Machine Learning framework that integrates behavioral feature extraction, deep learning analysis, and real-time mitigation mechanisms for comprehensive ransomware defense.

III. ANALYSIS AND DISCUSSION

Ransomware has become one of the most severe cybersecurity challenges, affecting both individuals and organizations worldwide. The proposed Python-based Machine Learning (ML) framework aims to detect, prevent, and mitigate

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. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141104

ransomware attacks through real-time behavioral monitoring and intelligent classification. This section analyzes the data used, evaluates the ML models, and discusses the system's performance and practical outcomes.

IV. PROPOSED SYSTEM

The proposed system aims to build an intelligent, Python-based Machine Learning framework capable of detecting, preventing, and mitigating ransomware attacks in real time. The system focuses on behavioral analysis rather than signature matching, ensuring effective detection of both known and unknown (zero-day) ransomware variants.

A. System Overview

The system continuously monitors the host environment and identifies abnormal activities that resemble ransomware behavior. It extracts real-time system metrics such as file access rate, entropy, process creation, and CPU/memory utilization. These behavioral patterns are then analyzed by trained ML models to determine whether the activity is benign or malicious**.**

If malicious behavior is detected, the system automatically initiates mitigation actions such as process termination**,** file quarantine**,** and backup restoration**.**

B. System Architecture

The overall architecture of the proposed ransomware-detection system is composed of the following modules:

1. **Data Collection Module**
   - Uses Python libraries (os, psutil, watchdog) to capture live system and file-level activities.
   - Gathers metrics such as file entropy, file size, I/O frequency, and process execution patterns.
   - Both normal and ransomware samples are collected to train the ML model.
2. **Feature Extraction and Preprocessing Module**
   - Extracts key behavioral features:
     - File entropy
     - File rename and extension changes
     - CPU/memory spikes
     - File-write frequency
   - Data is cleaned, normalized, and encoded using Pandas**,** NumPy**,** and Scikit-learn preprocessing tools.
3. **Machine Learning Model Training Module**
   - Trains and evaluates algorithms such as Random Forest**,** SVM**,** and CNN**.**
   - Performs hyperparameter tuning with Grid Search CV to maximize accuracy.
   - The trained model is serialized using joblib for integration into the real-time system.
4. **Detection and Classification Module**
   - In the deployed environment, new behavioral data is passed to the trained model.
   - If the prediction score exceeds a defined threshold, the process is classified as ransomware.
5. **Mitigation and Response Module**
   - Immediately executes countermeasures through automated scripts:
     - Terminates suspicious processes (psutil.Process().terminate())
     - Moves suspicious files to a quarantine folder
     - Restores original files from secure backup
     - Sends alerts to the administrator via email or log entries
6. **Visualization and Reporting Module**
   - Generates real-time dashboards using Matplotlib or Streamlit**.**
   - Displays detection trends, CPU usage, entropy changes, and response actions.

V. METHODOLOGY

1. **Data Acquisition:**
   Collection of normal and ransomware-infected file activity logs from publicly available sources (Kaggle, VirusShare) and controlled system simulations. Behavioral parameters such as file-write rate, entropy, CPU/memory usage, and file extension changes are recorded using Python libraries like psutil and watchdog.
2. **Data Preprocessing:**
   Cleaning and normalization of raw data using Pandas and NumPy. Missing or noisy values are handled, and important behavioral attributes are encoded into numerical features suitable for ML model training.
3. **Feature Engineering:**
   Extraction of key indicators of ransomware activity, including entropy variation, abnormal I/O patterns, rapid file encryption, and process spikes. Correlation analysis is performed to select the most relevant attributes.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141104

4. **Model Design and Training:**
   Implementation of Random Forest, Support Vector Machine (SVM), and Logistic Regression algorithms using Scikit-learn. Models are trained on labeled datasets and evaluated based on accuracy, precision, recall, and F1-score to identify the best-performing classifier.
5. **Real-Time Detection and Mitigation:**
   Integration of the trained ML model with a real-time monitoring system using the Watchdog library. When ransomware-like behavior is detected, the system automatically terminates the malicious process, quarantines affected files, and restores clean backups.
6. **Visualization and Alert System:**
   Real-time dashboards are developed using Streamlit and Matplotlib to display detection trends, system performance, and alerts to administrators for immediate response.

VI. RESULTS AND DISCUSSION

The proposed Python-based Machine Learning framework for ransomware detection and prevention was rigorously evaluated to determine its effectiveness, accuracy, and robustness in real-time environments. The experimental study focused on assessing the system's ability to correctly classify ransomware behavior, detect early-stage encryption activities, and initiate timely mitigation responses.
This section presents a comprehensive analysis of experimental outcomes, model comparisons, behavioral insights, and real-world applicability.

A. Experimental Setup

The ransomware detection system was implemented in Python 3.10 using Scikit-learn, Pandas, NumPy, and Matplotlib libraries. Experiments were conducted on a Windows 11 / Ubuntu 22.04 environment with an Intel i7 processor (16GB RAM).

Datasets Used:

- **Ransomware Activity Dataset:** Custom dataset created from controlled ransomware simulations (e.g., WannaCry, Locky, Cerber variants).
- **Normal File Activity Dataset:** System logs from regular user operations such as document editing, file copying, and browsing.
- **Combined Dataset:** Approximately 12,000 samples (6,000 ransomware behaviors and 6,000 benign samples).

Each sample contained multiple behavioral attributes, including:

- File-write rate
- File entropy
- CPU utilization
- Process creation frequency
- File extension changes

Data was split into 80% training and 20% testing using stratified sampling to maintain class balance. The Random Forest, SVM, and Logistic Regression models were trained and tuned using Grid Search for optimal performance.

B. Model Performance Evaluation

To evaluate detection capability, the models were compared based on Accuracy, Precision, Recall, F1-Score, and ROC-AUC values. These metrics collectively assess correctness, sensitivity, and reliability of the ransomware classification system.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	89.6	88.2	89.1	88.6
Support Vector Machine (SVM)	93.2	92.7	92.4	92.5
Random Forest Classifier	96.4	96.8	96.1	96.4

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141104

Key Observations:

- The Random Forest Classifier achieved the highest performance, accurately identifying ransomware activity patterns based on behavioral indicators.
- The model's ability to handle nonlinear data and perform ensemble voting helped minimize false positives.
- The average detection latency was 1.8 seconds, enabling near real-time response.

C. Detection and Mitigation Results

The system was tested under simulated ransomware attacks using file encryption scripts.
Results demonstrated the following outcomes:

- **Detection Rate:** 96% of ransomware activities were correctly detected.
- **False Positives:** Less than 3%, ensuring normal applications were not disrupted.
- **Automatic Response:** Suspicious processes were terminated instantly using psutil, and affected files were quarantined.
- **Backup Recovery:** Clean backups were successfully restored, preventing permanent data loss.

D. Visualization and Interpretability

- The Confusion Matrix revealed strong detection accuracy with minimal misclassification between benign and malicious classes.
- Feature Importance Analysis (via Random Forest) showed that:
  - File entropy
  - Write frequency
  - CPU spikes

  were the top predictors of ransomware behavior.
- Integration with LIME/SHAP provided interpretable visual explanations, highlighting how specific behavioral anomalies influenced classification results.

VII. CONCLUSION AND FUTURE SCOPE

A. Conclusion

Ransomware continues to pose a critical threat to cybersecurity, with attacks growing in frequency, sophistication, and financial impact. This study presented a Python-based Machine Learning framework for ransomware detection, mitigation, and prevention through behavioral analysis and intelligent classification**.**

The system successfully identifies ransomware activity in its early stages by analyzing process behaviors such as file entropy, CPU utilization, and file access frequency. Among all tested algorithms, the Random Forest Classifier achieved the best performance with 96.4% accuracy**,** 96.8% precision**,** and minimal false-positive rates.

The framework integrates real-time monitoring, automated process termination, and data recovery mechanisms to minimize damage during attacks. Moreover, the inclusion of Explainable AI (LIME/SHAP) enhances model transparency, making the system interpretable for cybersecurity analysts and IT professionals.

Overall, the results confirm that Machine Learning-driven behavioral detection provides a more adaptive, proactive, and robust approach than traditional signature-based antivirus systems. The developed model demonstrates high reliability, interpretability, and real-world practicality for defending against evolving ransomware threats.

B. Future Scope

While the proposed framework delivers promising results, several enhancements can further strengthen its efficiency and scalability:

1. **Deep Learning Integration:**
   Future versions can employ deep neural architectures such as LSTM or Autoencoders to improve temporal behavior learning and zero-day attack detection.
2. **Federated Learning Approach:**
   Implementing federated or distributed ML can enable multiple organizations to collaboratively train models without sharing sensitive data, improving global ransomware resilience.
3. **Cloud-based Threat Intelligence:**
   Integration with cloud-based intelligence feeds (e.g., VirusTotal, CISA) could enhance real-time threat updates and improve adaptability to new ransomware variants.
4. **Cross-Platform Compatibility:**
   Expanding the framework to support Windows, Linux, and Android environments will broaden its usability in enterprise and personal systems.
5. **Automated Incident Response System:**
   Coupling the detection model with SOAR (Security Orchestration, Automation, and Response) tools can enable automatic isolation, backup restoration, and recovery after an attack.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141104

6. **User Awareness Module:**
   Developing an interactive dashboard that alerts users, logs incidents, and provides educational insights could increase cybersecurity awareness and minimize user-triggered infections.

REFERENCES

[1] M. Sgandurra, L. Munoz-Gonzalez, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations, and use for detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 285–299, 2020.
[2] Y. K. Mehra, A. Sharma, and S. Singh, "Ransomware detection using machine learning techniques: A comparative analysis," *International Journal of Information Security Science*, vol. 13, no. 4, pp. 101–112, 2023.
[3] M. Kharrazi, R. Rahman, and T. Ahmad, "A behavior-based ransomware detection system using Random Forest and Gradient Boosting," *Journal of Cybersecurity and Digital Forensics*, vol. 5, no. 1, pp. 45–59, 2022.
[4] H. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluating deep learning approaches to characterize and classify ransomware," *Procedia Computer Science*, vol. 132, pp. 1028–1037, 2023.
[5] A. Mohaisen and O. Alrawi, "Unveiling ransomware evolution: Analysis and detection using machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–36, 2024.
[6] S. Agrawal and M. Tapaswi, "Next-generation ransomware detection using hybrid deep learning and explainable AI," *IEEE Access*, vol. 12, pp. 84127–84140, 2024.
[7] R. Hasan, J. Rahman, and S. Huda, "Explainable artificial intelligence for malware and ransomware detection: A survey," *Computers & Security*, vol. 140, p. 103567, 2024.
[8] A. K. Jain and A. Gupta, "Real-time ransomware detection using Python-based machine learning framework," *International Journal of Emerging Technologies in Computer Science & Electronics (IJETCSE)*, vol. 36, no. 5, pp. 87–94, 2023.
[9] N. Singh and P. Kaur, "Behavioral analysis-based ransomware prevention system using Random Forest and LSTM," *International Conference on Advances in Cybersecurity (ICACS)*, 2024.
[10] U. N. Kumar, "Machine learning approaches for detecting ransomware activity through system behavior," *Journal of Information Assurance and Security*, vol. 19, no. 1, pp. 55–63, 2025.

