AI-Based Early Detection and Mitigation of Ransomware Attacks Using Behavioural Analysis

Yajuvendra Singh[1], Anamika Agarwal[2], Rajveer Shukla[3]
[1,2,3]*School Of Computer Application, Babu Banarasi Das University, Lucknow, India*
doi.org/10.64643/IJIRT209700

***Abstract*—Ransomware is a malicious software type which prevents users from accessing files or systems and demands payment from them. While signature-based defense mechanisms can be effective against known threats, they become less effective against new or modified variants since the signature for such variants is unknown. This paper outlines the design of an AI-based system for detecting ransomware early on and safely mitigating its effects using behavioural analysis. The proposed system collects behavioural data (i.e., file operations, process activity, system calls, network behavior, registry modifications, CPU and memory utilization, and file entropy), processes it, selects the most informative features, and develops machine-learning models (Decision Tree, Random Forest, SVM, Logistic Regression, Gradient Boosting, XGBoost and Neural Networks) to categorize activity as Normal, Suspicious or Potential Ransomware. When highly suspicious activity is identified, the system provides early warnings and safe, controlled responses within a laboratory setting. This paper describes the problem formulation, methodology, evaluation plan and anticipated results of the proposed study. It is a research proposal; therefore, no experimental results are presented.**

***Index Terms*—ransomware detection, behavioural analysis, machine learning, early warning, cybersecurity, malware, anomaly detection, file entropy, controlled mitigation, false-positive reduction**

I. INTRODUCTION

Ransomware has become one of the major cybersecurity challenges for modern computing environments. It is malicious software that interferes with access to files, applications or entire systems and may demand payment from the victim [6], [7]. Individuals, educational institutions, businesses, healthcare organisations, financial institutions and government departments are all potential targets.

Conventional security tools rely largely on predefined signatures, rules and previously identified malware patterns. These are effective against known threats, but their effectiveness decreases when attackers release new variants or modify existing programs to evade detection.

Behavioural analysis offers a different view. Instead of asking what a program is, it observes what a process does. Ransomware tends to show rapid file modification, repeated file access, unusual file renaming, abnormal process execution, shifts in resource usage and suspicious network communication [1]– [3]. Artificial Intelligence (AI) and Machine Learning (ML) can learn the relationship between such activity and ransomware from labelled benign and malicious data [5].

This paper proposes an AI-based framework for early ransomware detection and controlled mitigation using behavioural analysis. The work covers behavioural feature identification, data preprocessing, feature selection, ML model development, comparative evaluation, early-warning generation and controlled mitigation. The aim is to identify suspicious behaviour early, so that security action can begin before extensive disruption occurs.

The remainder of the paper is organised as follows. Section II gives the motivation, Section III reviews related work, Section IV identifies research gaps, Section V formulates the problem, Section VI states the objectives, Section VII describes the proposed methodology, Section VIII gives the evaluation plan, Section IX lists expected outcomes, and Section X concludes.

II. MOTIVATION

Ransomware attacks are increasing in frequency and severity and are becoming increasingly difficult to detect due to the constant evolution of attacker tactics. Conventional signature-based tools may fail to recognize new or modified samples of ransomware.

The primary motivation for conducting this research is to develop a tool that identifies ransomware based upon its behavior rather than its signature. During an attack ransomware may modify or rename many files, repeatedly access files, create suspicious processes or contact external systems. ML can analyze these activities and identify patterns associated with ransomware, allowing for early detection. A secondary motivation for conducting this research is reducing false positives. Legitimate applications such as backup applications, compression applications or indexing applications also generate significant amounts of file and system activity.

Therefore, selecting appropriate behavioral features and evaluating various algorithms is critical for improving the accuracy of detection. A third motivation for conducting this research is linking detection with response. Once highly suspicious activity is identified, the system should generate an alert, log the activity and enable controlled mitigation within a secure research environment.

III. RELATED WORK

Ransomware detection has been studied through behavioural analysis, file-system monitoring, dynamic analysis and machine learning. Fig. 1 summarises these approaches and how they feed detection and response.

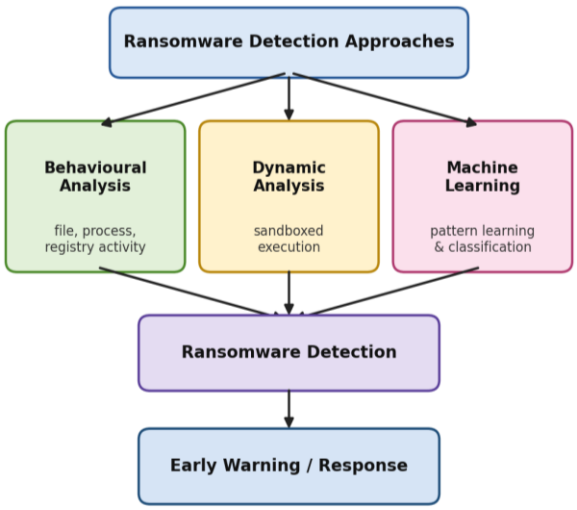

Fig. 1. Major approaches used in ransomware detection.

The flowchart combines behavioural analysis, dynamic analysis, and machine learning to detect ransomware, identify suspicious activities, and generate early warnings.

Kharraz et al. [1] developed UNVEIL, an automated method that detects ransomware from suspicious file-system behavior demonstrating the utility of monitoring file activity in addition to relying solely on signatures. Scaife et al. [2] developed CryptoLock which monitors user data and file-system activity to identify and prevent ransomware from executing demonstrating that abnormal file operations are a useful indicator of ransomware. Continella et al. [3] developed ShieldFS, a ransomware-aware file system that detects and mitigates malicious file modifications to protect user data.

Sgandurra et al. [4] studied automated dynamic analysis of ransomware and demonstrated that behavior exhibited during execution can assist automated detection. Kolodenker et al. [5] developed EldeRan which combines behavioral features with machine learning to classify ransomware. Richardson and North [6] and Brewer [7] provide general overviews of ransomware evolution, mitigation, detection and prevention. Table I summarises the key contributions and the remaining limitations.

TABLE I. SUMMARY OF RELATED WORK

Ref.	System / Study	Main Idea	Observation
[1]	UNVEIL (2016)	Automated detection from suspicious file-system behaviour	Stresses file activity over signatures
[2]	CryptoLock (2016)	Monitors user data and file operations to stop attacks	Abnormal file operations indicate ransomware
[3]	ShieldFS (2016)	Ransomware-aware file system; detects and mitigates modifications	Protects user data via file-system monitoring
[4]	Dynamic analysis (2016)	Studies ransomware behaviour during execution	Dynamic behaviour aids automated detection
[5]	EldeRan (2017)	Behavioural features with ML classification	Shows behaviour suits ML classification

Ref.	System / Study	Main Idea	Observation
[6], [7]	Surveys (2016–2017)	Overviews of evolution, detection, prevention and mitigation	Context on threat and defence

However, early detection, reduction of false positives, identification of relevant behavioural features and integration of detection with mitigation remain unresolved issues motivating the current study.

IV. RESEARCH GAPS

Several areas need further attention:

- New or modified variants: many approaches are tested on known samples, so it is unclear whether behavioural features generalise to unseen or altered ransomware.
- False positives: benign programs can produce file and system activity that resembles an attack; feature selection and model tuning are needed to separate them.
- Early detection: detecting an attack after many files are affected reduces the value of a defence; indicators must be found earlier.
- Detection–mitigation integration: prior work has largely concentrated on detection, whereas alerting, logging and controlled response can be combined in one framework.
- Model comparison: algorithms behave differently depending on the dataset and features, so a comparative evaluation is required.

Based on these gaps, this work proposes an integrated framework combining behavioural data collection, feature selection, ML-based detection, early warning and controlled mitigation in a safe research environment.

V. PROBLEM FORMULATION

Ransomware can perform numerous file, process, system and network operations in a relatively short time frame; furthermore signature-based methods may not recognize new or modified samples of ransomware. The research question addressed by this study is: how can AI and ML techniques be used to analyze behavioral characteristics of computer-system activity for early ransomware detection and controlled mitigation while maintaining high levels of detection accuracy and low false-positive rates?

Formally, let x(t) be the feature vector computed over a time window ending at t. A trained model f produces a ransomware score s(t) = f(x(t)) in [0, 1]. The activity is assigned to one of three classes using two thresholds $T1 < T2$ (Fig. 2): Normal if $s < T1$, Suspicious if $T1 \leq s < T2$, and Potential Ransomware if $s \geq T2$. The thresholds are tuned to balance early warning against false alarms.

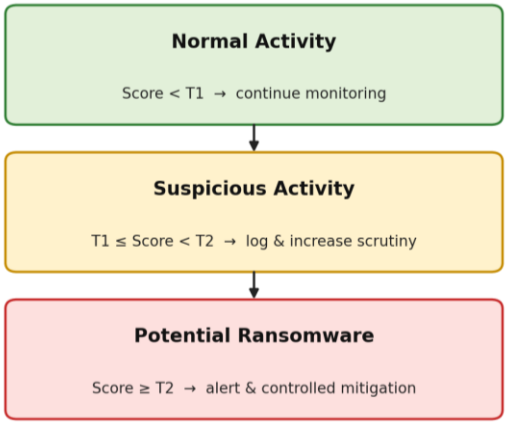

Fig. 2. Ransomware activity classification process.

The flowchart classifies activities using risk thresholds: normal activities are monitored, suspicious activities logged, and potential ransomware triggers alerts and mitigation.

The behavioural characteristics to be investigated are grouped in Fig. 3 and listed in Table II.

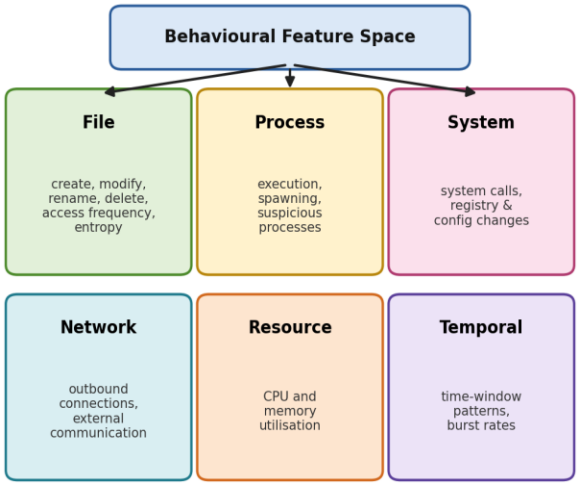

Fig. 3. Behavioural feature categories.

The flowchart illustrates six behavioural feature categories: file, process, system, network, resource,

and temporal activities, helping identify suspicious ransomware behaviour.

TABLE II. BEHAVIOURAL CHARACTERISTICS UNDER STUDY

Category	Characteristic	Why it matters
File	Creation and modification	Mass changes to user files are typical of encryption
File	Renaming and deletion	Extension changes and removal of originals
File	Access frequency	Repeated reads and writes in short time
File	Entropy	Encrypted content has high randomness [11]
Process	Execution behaviour	Unusual or suspicious process creation
System	System-call patterns	Low-level view of program activity
System	Registry / configuration changes	Persistence and system tampering
Resource	CPU and memory utilisation	Resource shifts during intensive operations
Network	Network-related activity	Communication with external systems
Temporal	Temporal patterns	Burst behaviour within time windows

VI. OBJECTIVES

The objectives of the proposed research are:

- To study ransomware attacks and existing behavioural-analysis techniques for detection.
- To identify and analyse behavioural characteristics that distinguish ransomware from normal system behaviour.
- To collect, preprocess and select relevant behavioural features from suitable datasets containing benign and ransomware-related activity.
- To develop and compare ML models for classifying activity as Normal, Suspicious or Potential Ransomware.
- To study decision thresholds that enable early warning with a low false-positive rate.

To design an early-warning and controlled-mitigation mechanism and evaluate it in a safe laboratory environment.

VII. PROPOSED METHODOLOGY

The research proceeds through the stages shown in Fig. 4, described below

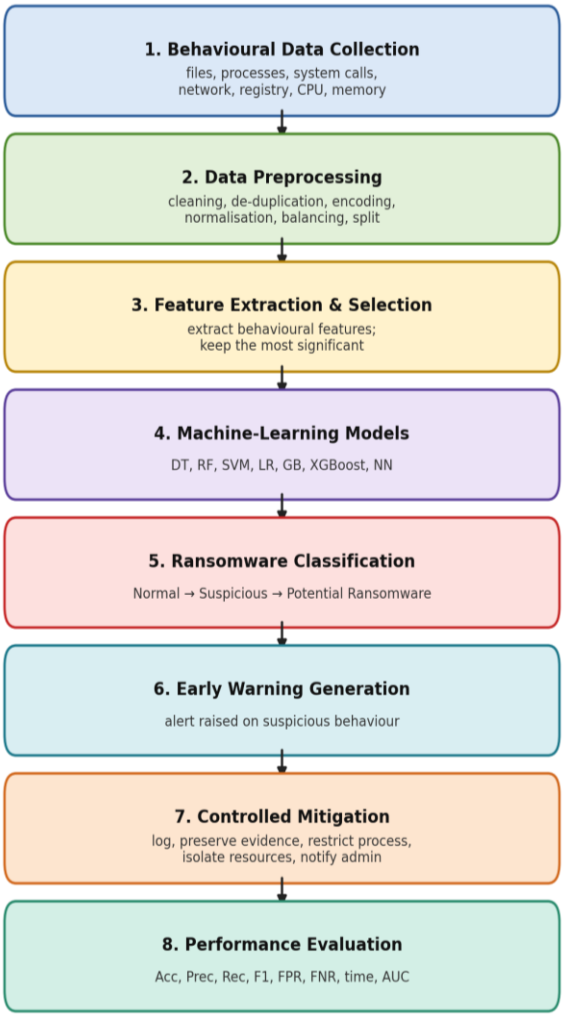

Fig. 4. Proposed framework for early ransomware detection and mitigation

The flowchart illustrates data collection, preprocessing, feature selection, machine learning classification, early warning, controlled mitigation, and performance evaluation for ransomware detection.

A. Literature Review

Papers on ransomware, behavioural analysis, malware and anomaly detection, AI/ML and mitigation are reviewed to establish existing approaches and gaps.

B. Dataset Selection

A suitable publicly available cybersecurity dataset with benign and ransomware or malicious behavioural data will be selected. It may include file operations, process activity, system calls, network behaviour, registry modifications, CPU and memory usage, access patterns and file entropy.

C. Data Preprocessing

The data will be cleaned by removing duplicates and handling missing values. Categorical variables will be encoded and numerical features normalised. Class imbalance will be addressed, for example with resampling such as SMOTE [12], and the data split into training and testing subsets.

D. Feature Extraction and Selection

Behavioural features are extracted and feature-selection techniques applied to retain the most significant ones. This removes irrelevant information, reduces computational cost, improves efficiency and can improve detection performance.

E. Model Development, Training and Testing

The algorithms in Table III will be implemented, for example using scikit-learn [13] and XGBoost [10]. Training data develops the models and held-out testing data measures performance on unseen observations.

TABLE III. CANDIDATE MACHINE-LEARNING MODELS

Model	Role in the study
Decision Tree	Interpretable baseline classifier
Random Forest [8]	Ensemble of trees; robust to noisy features
Support Vector Machine [9]	Margin-based classifier for separating classes
Logistic Regression	Linear baseline producing class probabilities
Gradient Boosting / XGBoost [10]	Boosted tree ensembles for tabular data
Neural Network	Learns non-linear patterns, where appropriate

F. Ransomware Detection

The trained models classify activity as Normal → Suspicious → Potential Ransomware (Fig. 2). The decision thresholds are studied so that early identification and warning of potentially malicious behaviour is possible.

G. Early Warning and Controlled Mitigation

When highly suspicious behaviour is detected, the framework investigates safe response mechanisms (Table IV). All mitigation experiments are conducted inside a controlled research or laboratory environment.

TABLE IV. TIERED RESPONSE ACTIONS

Class	Response
Normal	Continue monitoring; no action
Suspicious	Log activity; preserve system logs; monitor the process more closely
Potential Ransomware	Generate security alert; restrict suspicious process; isolate test resources; notify administrator

VIII. PERFORMANCE EVALUATION PLAN

Models will be assessed using metrics listed in Table 5 (where applicable) using held-out test data sets; models will be compared to determine which offers superior performance. ROC analysis will follow Fawcett [14].

Of particular importance will be assessment of detection time since the objective is to alert users prior to substantial damage occurring.

TABLE V. EVALUATION METRICS

Metric	Definition / purpose
Accuracy	(TP + TN) / (TP + TN + FP + FN)
Precision	TP / (TP + FP); reliability of alerts
Recall	TP / (TP + FN); share of ransomware detected
F1-Score	Harmonic mean of precision and recall
FPR	FP / (FP + TN); false alarms on benign activity
FNR	FN / (FN + TP); missed ransomware
Detection time	Delay from first malicious activity to alert
ROC-AUC	Threshold-independent separability of classes

IX. EXPECTED OUTCOMES

- An AI-based ransomware detection framework built on behavioural analysis of system activity.
- Identification of important behavioural features separating ransomware from normal behaviour.
- Development and comparison of ML models for detecting and classifying ransomware-related activity.
- Improved early detection by identifying suspicious activity before significant damage occurs.
- Fewer false positives through suitable feature selection and model optimisation.
- An early-warning and controlled-mitigation mechanism that generates alerts and supports safe responses.

Limitations

This paper is a proposal; the expected outcomes are objectives of the work, not measured results. Findings will depend on the chosen dataset, and behaviour in a laboratory may differ from real deployments.

X. CONCLUSION

This paper proposed an AI-based framework for early ransomware detection/controlled mitigation using behavioral analysis. By incorporating behavioral data collection/feature selection/comparative ML modeling/three-level classification/early warning/safe response mechanisms into a single framework this study addresses gaps related to detection of variants/false positives/early detection/detection-mitigation integration. Future studies will implement this framework on a selected dataset(s)/report on metrics presented in Table 5 evaluate generalizability to new variants of ransomware.

REFERENCES

[1] Kharraz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware," in *Proc. 25th USENIX Security Symp.*, 2016.

[2] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and drop it): Stopping ransomware attacks on user data," in *Proc. 36th IEEE Int. Conf. Distributed Computing Syst. (ICDCS)*, 2016.

[3] A. Continella *et al*., "ShieldFS: The last word in ransomware resiliency," in *Proc. 32nd Annu. Computer Security Applications Conf. (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," *arXiv:1609.03020*, 2016.

[5] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele, "EldeRan: A machine learning approach to ransomware detection," in *Proc. 13th Int. Conf. Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2017.

[6] R. Richardson and M. North, "Ransomware: Evolution, mitigation and prevention," *Int. Manage. Rev.*, vol. 13, no. 1, pp. 10–21, 2017.

[7] R. Brewer, "Ransomware attacks: Detection, prevention and cure," *Network Security*, vol. 2016, no. 9, pp. 5–9, 2016.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[11] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[13] F. Pedregosa *et al*., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[14] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[15] A. Agarwal, S. B. Verma, B. K. Gupta, S. Gupta, and S. Verma, "Ransomware anatomy, impact and mitigation strategies."