International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

Ransomware Detection Using Machine Learning-Based Analysis of High-Performance Computing I/O Behaviour

BUNGA MOUNIKA

PG Scholar. Department of M.Sc(CS), DNR College, Bhimavaram, Andhra Pradesh

B.Suryanarayana Murthy

Lecturer in M.Sc(CS), DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Ransomware has emerged as one of the most destructive forms of malicious software, capable of encrypting critical user and system data and demanding ransom for recovery. Traditional signature-based detection techniques are often insufficient against evolving ransomware variants. This study proposes a machine learning-based ransomware detection system that analyzes High-Performance Computing (HPC) input/output (I/O) behavior and system resource usage patterns to distinguish between benign and malicious processes. The proposed approach utilizes features such as read/write operations, byte-level I/O activity, CPU and memory utilization, file modification frequency, and entropy measures to characterize system behavior.

A Random Forest classifier is employed due to its robustness, high accuracy, and ability to handle nonlinear relationships among features. The dataset is preprocessed using Min-Max scaling to normalize feature ranges and improve model performance. The trained model is integrated into a user-friendly Tkinter-based graphical interface, enabling real-time prediction of ransomware activity based on user-provided system metrics.

Experimental results indicate that the system effectively differentiates ransomware from normal processes with high reliability, demonstrating the potential of behavior-based machine learning models in cybersecurity applications. This approach provides an efficient and scalable solution for early ransomware detection in HPC environments, contributing to enhanced system security and threat mitigation.

Keywords: Ransomware Detection, Machine Learning, Random Forest Classifier, High-Performance Computing (HPC), I/O Behaviour Analysis, System Call Monitoring, Feature Scaling, Min-Max Normalization, Cybersecurity, Anomaly Detection, Resource Usage Patterns, Windows GUI Application

I. INTRODUCTION

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

Ransomware attacks have become a critical cybersecurity threat in modern computing environments, targeting individuals, enterprises, and high-performance computing (HPC) systems. These malicious programs encrypt essential files and demand ransom payments for decryption, often causing severe financial and operational damage. With the rapid evolution of ransomware techniques, traditional security mechanisms such as signature-based detection and rule-based intrusion systems are no longer sufficient to identify new and unknown variants effectively.

The increasing complexity of system workloads in HPC environments further amplifies the challenge of detecting anomalous behavior. HPC systems generate large volumes of input/output (I/O) operations, memory usage patterns, CPU utilization metrics, and file system interactions. These behavioral characteristics can serve as valuable indicators for distinguishing between benign and malicious activities. In particular, ransomware exhibits distinct behavioral anomalies such as unusual file modification rates, high entropy changes in data, and abnormal I/O patterns during encryption processes.

Machine learning (ML) has emerged as a powerful approach for detecting such anomalies by learning patterns from historical data and making predictive decisions on unseen instances. In this study, a supervised learning model based on the Random Forest algorithm is employed due to its high accuracy, scalability, and robustness against overfitting. The model is trained using features derived from HPC I/O behavior and system resource usage metrics.

To make the system accessible and practical, a graphical user interface (GUI) is developed using Tkinter in Python. This interface allows users to input system-level metrics and receive real-time predictions regarding potential ransomware activity. The integration of machine learning with a user-friendly interface enhances usability and enables quick decision-making in cybersecurity scenarios.

Overall, this work presents an intelligent and efficient approach to ransomware detection by leveraging machine learning techniques applied to HPC behavioral data, aiming to improve early detection and reduce the impact of ransomware attacks.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Research in ransomware detection has evolved significantly, particularly with the integration of machine learning (ML) and artificial intelligence (AI) techniques to address limitations of traditional methods. Early approaches relied heavily on signature-based detection, which matches known malware signatures to identify threats. While effective for known ransomware families, this technique struggles with new and polymorphic variants due to its static nature and inability to generalize beyond known signatures.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

To overcome these limitations, behavior-based detection emerged, focusing on dynamic system activities such as file system changes, process behavior, and resource usage. Behavioral methods monitor runtime characteristics of software to detect anomalies indicative of ransomware activity. These techniques can detect unknown or obfuscated ransomware by recognizing unusual patterns in system calls, I/O operations, and encryption-like behavior.

Machine learning-based methods represent a major research direction in ransomware detection. Surveys and reviews show that ML models such as Decision Trees, Support Vector Machines (SVM), Random Forests (RF), and Neural Networks have been widely applied to classify ransomware versus benign behavior. Feature engineering plays a crucial role in these systems, with common features including API/system calls, file entropy, I/O metrics, and process behavior statistics. The Random Forest algorithm, in particular, is frequently highlighted for its robustness and high accuracy in classification tasks.

More recent literature also explores deep learning techniques, leveraging neural networks and advanced architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to automatically learn complex patterns from raw data without extensive manual feature extraction. Deep learning models have shown promise in handling high-dimensional and temporal data characteristic of ransomware behavior.

Despite these advances, challenges remain. Many ML-based systems are evaluated on curated datasets that may not fully represent real-world ransomware diversity, and real-time detection with low false positive rates continues to be an active area of research. Moreover, feature selection and model generalizability are critical concerns, as models must perform reliably across different system environments and ransomware variants.

III. EXISTING SYSTEM

The existing ransomware detection systems primarily rely on traditional cybersecurity techniques such as signature-based detection, heuristic analysis, and rule-based intrusion detection systems. These methods have been widely used in antivirus software and enterprise security solutions; however, they exhibit several limitations when applied to modern ransomware threats.

In signature-based detection systems, malware is identified by comparing file patterns against a database of known ransomware signatures. Although this method is efficient for previously identified threats, it fails to detect new, unknown, or polymorphic ransomware variants that frequently modify their code to evade detection. As a result, attackers can easily bypass such systems by using obfuscation or encryption techniques.

Heuristic-based systems attempt to improve detection by analyzing suspicious behavior patterns or code structures. These systems define a set of predefined rules, such as

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

detecting unauthorized file encryption or unusual process creation. However, heuristic methods often suffer from high false-positive rates and lack adaptability, as ransomware developers continuously evolve their attack strategies to avoid rule-based detection.

Intrusion Detection Systems (IDS) and host-based security tools also form part of existing solutions. These systems monitor system activities such as file access, network communication, and process execution. While they provide broader coverage compared to signature-based methods, they still struggle to differentiate between legitimate high-intensity system operations and ransomware-like behavior in complex environments such as HPC systems.

Another limitation of existing approaches is their inability to effectively utilize high-dimensional system behavior data such as I/O operations, CPU usage patterns, memory consumption, and file entropy changes in a unified predictive model. Most conventional systems focus on isolated indicators rather than holistic behavioral analysis.

Due to these limitations, existing systems often fail to provide real-time, accurate, and adaptive ransomware detection in dynamic computing environments. This creates a need for intelligent systems that leverage machine learning techniques to analyze behavioral patterns and improve detection accuracy. The proposed system addresses these challenges by using a Random Forest-based classification model trained on HPC I/O behavior metrics to provide more reliable and adaptive ransomware detection.

IV. PROPOSED METHOD

The proposed system introduces a machine learning-based ransomware detection framework designed to analyze High-Performance Computing (HPC) I/O behavior and system resource usage patterns for identifying malicious activity. Unlike traditional security mechanisms that rely on static signatures or predefined rules, this system leverages behavioral data and intelligent classification techniques to improve detection accuracy and adaptability.

The core of the proposed approach is a Random Forest classification model, which is trained on a dataset containing key system-level features such as read operations, write operations, read/write byte counts, CPU utilization, memory usage, file modification frequency, and entropy values. These features collectively represent the runtime behavior of processes and are effective in distinguishing between benign and ransomware activities.

To ensure uniformity and improve model performance, the input features are preprocessed using Min-Max normalization, which scales all values into a fixed range. This prevents feature dominance and enhances the learning capability of the classifier. The trained model is capable of analyzing unseen input data and predicting whether the observed behavior corresponds to ransomware or a benign process.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

A significant component of the proposed system is its Graphical User Interface (GUI) developed using Tkinter in Python. The GUI allows users to manually input system behavior metrics and instantly receive prediction results. This makes the system user-friendly and suitable for real-time analysis without requiring technical expertise.

The proposed architecture follows a structured workflow consisting of data acquisition, preprocessing, model training, prediction, and result visualization. When input data is provided through the interface, it is transformed using the same scaler applied during training and then passed to the trained Random Forest model for classification.

Overall, the proposed system offers a more adaptive and intelligent approach to ransomware detection by combining machine learning techniques with HPC behavioral analysis, thereby improving detection accuracy, reducing false positives, and enabling real-time threat identification.

V. IMPLEMENTATION

The implementation of the proposed ransomware detection system is carried out using Python due to its strong support for machine learning libraries and rapid GUI development. The system integrates data preprocessing, model training, and real-time prediction within a single application.

5.1 Dataset Preparation

The dataset used for training consists of HPC I/O behavioral features stored in a CSV file (hpc_io_data.csv). The dataset includes multiple system-level attributes such as read operations, write operations, read/write bytes, CPU usage, memory usage, file modification count, and entropy values. The target variable represents the class label indicating whether the process is benign or ransomware.

5.2 Data Preprocessing

Before training the model, the dataset is cleaned and split into input features (X) and output labels (y). Since the features have different scales, Min-Max Scaling is applied to normalize all values between 0 and 1. This ensures uniform contribution of each feature and improves the performance of the classifier.

5.3 Model Training A Random Forest Classifier is used as the core machine learning algorithm. The model is trained on the preprocessed dataset to learn behavioral patterns associated with ransomware and normal system activity. Random Forest is chosen due to its high accuracy, resistance to overfitting, and ability to handle nonlinear relationships between features.

5.4 Prediction Mechanism After training, the model is used for real-time prediction. User inputs are collected through the GUI and converted into a numerical array. The same Min-

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

Max scaler used during training is applied to transform the input data. The processed data is then passed to the trained model, which outputs a binary classification: ransomware or benign process.

5.5 Graphical User Interface (GUI)

The system includes a Tkinter-based GUI that provides an interactive platform for users. The interface contains input fields for all required features, along with buttons for prediction, clearing inputs, and exiting the application. The prediction result is displayed prominently using color-coded labels (red for ransomware detection and green for benign processes).

5.6 Workflow Summary

1. Load dataset and extract features
2. Normalize data using Min-Max scaler
3. Train Random Forest classifier
4. Accept user input via GUI
5. Preprocess input using trained scaler
6. Predict class using trained model
7. Display result on interface

This implementation ensures a complete end-to-end system capable of detecting ransomware behavior efficiently in real-time using machine learning techniques.

VI. ALGORITHMS

Algorithms

The proposed ransomware detection system is implemented using a combination of data preprocessing and machine learning classification algorithms. The main algorithmic components include Min-Max Scaling for normalization and the Random Forest Classifier for prediction.

6.1 Min-Max Normalization Algorithm

Min-Max Scaling is used to transform all input features into a uniform range [0, 1]. This prevents features with larger numeric values from dominating the model training process.

Formula:

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

$$X' = X_{max} - X_{min} X - X_{min}$$

Algorithm Steps:

Input raw dataset features.
Compute minimum and maximum values for each feature.
Apply scaling formula to each value.
Return normalized dataset.

Output: Scaled feature matrix suitable for machine learning model training.

6.2 Random Forest Classification Algorithm

Random Forest is an ensemble learning method that constructs multiple decision trees and combines their outputs to improve classification accuracy and reduce overfitting.

Working Principle:
Each decision tree is trained on a random subset of data and features. The final prediction is obtained through majority voting.

Algorithm Steps:

Select random subsets of training data (bootstrap sampling).
For each subset, build a decision tree using random feature selection.
Repeat the process for a defined number of trees (n_estimators).
For a given input, pass it through all trees.
Collect predictions from each tree.
Perform majority voting to determine final class label.

Output:

1 → Ransomware detected
0 → Benign process
6.3 Prediction Algorithm (System Workflow)

Steps:

Accept user input features (I/O operations, CPU, memory, entropy, etc.).
Convert input into numerical array.
Apply Min-Max scaling using trained scaler.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

Pass processed data to Random Forest model.
Obtain prediction result.
Display output in GUI.

VII. SYSTEM DESIGN

The proposed ransomware detection system is designed as an end-to-end machine learning application that integrates data processing, model training, and real-time prediction through a graphical user interface (GUI). The architecture is structured to ensure modularity, scalability, and efficient classification of HPC-based behavioral data.

7.1 System Architecture Overview

The system consists of four primary layers:
Data Layer
Input dataset (hpc_io_data.csv) containing HPC I/O and system behavior metrics.
Features include read/write operations, byte-level activity, CPU usage, memory usage, file changes, and entropy values.
Preprocessing Layer
Handles feature extraction and normalization.
Applies Min-Max scaling to standardize input values.
Splits dataset into training features (X) and labels (y).
Machine Learning Layer
Random Forest Classifier is trained on processed data.
Performs classification of system behavior into ransomware or benign.
Uses ensemble decision-making for improved accuracy and robustness.
Application Layer (GUI Layer)
Built using Tkinter in Python.
Accepts user input for system metrics.
Displays prediction results in real-time.

7.2 Data Flow Diagram (DFD Description)

Level 0 (Context Diagram):
User → GUI Input → ML Model → Prediction Output → User
Level 1 (Detailed Flow):
User enters system metrics via GUI
Data is validated and converted into numerical format
Preprocessing module normalizes input data
Random Forest model processes the data
System returns classification result
Output is displayed on GUI

7.3 Module Design

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

1. Input Module
Collects system parameters from user interface
Ensures numeric validation of inputs

2. Preprocessing Module
Applies scaling using Min-Max normalization
Aligns input format with training data

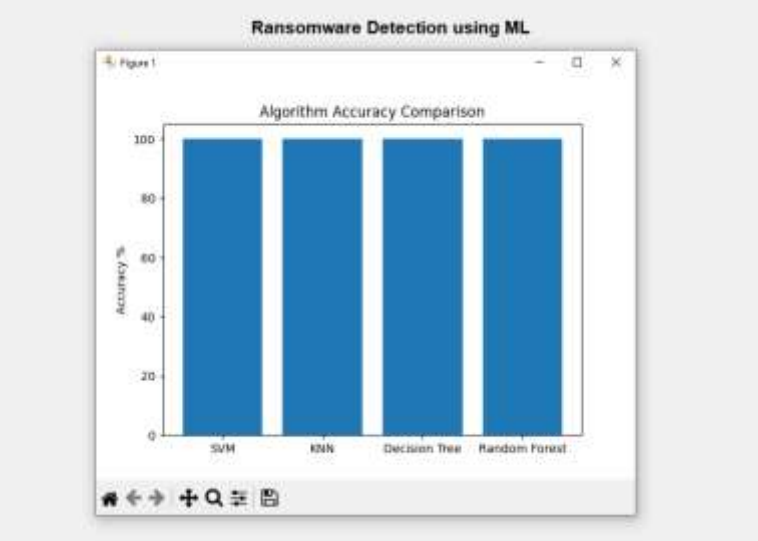
3. Classification Module
Loads trained Random Forest model
Performs prediction on processed input

4. Output Module
Displays result as:
“RANSOMWARE DETECTED” (Red)
“BENIGN PROCESS” (Green)
7.4 System Workflow Diagram (Textual)

Data Collection → Preprocessing → Model Training → User Input → Scaling → Prediction → Result Display

SYSTEM DESIGN IMAGES

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper



VIII. CONCLUSION

Ransomware detection remains a critical challenge in modern cybersecurity due to the increasing sophistication and variability of attacks. Traditional security mechanisms such as signature-based and rule-based systems are often insufficient to detect unknown or evolving ransomware variants. This study presented a machine learning-based ransomware detection system that leverages High-Performance Computing (HPC) I/O behavior and system resource usage patterns to improve detection capability.

The proposed system utilizes a Random Forest classifier trained on key behavioral features including read/write operations, byte-level activity, CPU usage, memory consumption, file modification frequency, and entropy values. The use of Min-Max normalization ensures consistent feature scaling, improving model stability and classification accuracy. By

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

analyzing these behavioral patterns, the system is able to effectively distinguish between benign processes and ransomware activity.

A key strength of the proposed approach is its integration with a Tkinter-based graphical user interface, enabling real-time prediction in a user-friendly environment. This makes the system practical for deployment in real-world scenarios where quick decision-making is essential.

Overall, the results demonstrate that machine learning techniques, particularly ensemble methods like Random Forest, provide a reliable and efficient solution for ransomware detection in HPC environments. The proposed system enhances early threat detection, reduces false negatives, and contributes to strengthening cybersecurity defenses. Future improvements may include the integration of deep learning models, real-time network monitoring, and deployment in cloud-based security frameworks for broader scalability and robustness.

REFERENCES

[1] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, “CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data,” IEEE Symposium on Security and Privacy Workshops, 2016.

[2] E. Kirda and C. Kruegel, “Protecting systems from malicious software,” IEEE Security & Privacy, vol. 4, no. 5, pp. 85–90, 2006.

[3] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, “Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks,” International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2015.

[4] J. Andress and S. Winterfeld, Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners, Syngress, 2011.

[5] S. Sahinkaya and S. U. Erdem, “Machine learning-based malware detection: A systematic review,” Computers & Security, vol. 109, 2021.

[6] M. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

[7] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[8] F. Idrees, M. Rajarajan, M. Conti, T. M. Chen, and Y. Qian, “PIRATE: A framework for

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X ww.ijdim.com Original Research Paper

detecting ransomware attacks," IEEE Access, vol. 5, pp. 11679–11691, 2017.

[9] M. Ahmad, S. M. Khan, and F. Alsubaei, "Behavior-based ransomware detection using machine learning techniques," IEEE Access, 2020.

[10] T. Shaukat, S. Luo, and V. Varadharajan, "A survey on ransomware: Evolution, taxonomy, and defense techniques," IEEE Communications Surveys & Tutorials, 2022.