**A REPORT**

**ON**

**AI-DRIVEN ADAPTIVE SECURITY AND RECOVERY PIPELINE**

**BITS ZG628T: Dissertation**

**BY**

**Dabhade Pooja Bhanudas**

**2023MT93089**

**AT**

**Dell Technologies, Bangalore**

****

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE**

**PILANI (RAJASTHAN)**

**(APRIL 2025)**

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI**

**(APRIL 2025)**

**A REPORT**

**ON**

**AI-DRIVEN ADAPTIVE SECURITY AND RECOVERY PIPELINE**

**BITS ZG628T: Dissertation**

**BY**

**Dabhade Pooja Bhanudas**

**2023MT93089**

**Prepared in partial fulfilment of the**

**WILP Dissertation Course**

**AT**

**Dell Technologies, Bangalore**

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who have supported and guided me throughout the course of this project. Their encouragement and invaluable contributions have played a pivotal role in the successful completion of this work.

First and foremost, I extend my heartfelt thanks to **Michael Dell**, Head of the Organization, for providing the opportunity to pursue higher education while working, thereby fostering an environment that encourages continuous learning and professional growth.

I am deeply grateful to my supervisor and additional examiner, **Rupam Kundu**, for his unwavering support, expert guidance, and invaluable insights that have greatly contributed to shaping this project. His mentorship and encouragement have been instrumental in overcoming challenges and refining my work.

I would also like to express my sincere appreciation to my managers, **Asif Khan** and **Preeti Varma**, for recognizing my potential and granting me this opportunity. Their continuous support, motivation, and understanding have provided the necessary foundation for the successful execution of this project.

Lastly, I extend my gratitude to all those, both within and outside the organization, who have directly or indirectly contributed to this endeavour. Their encouragement and assistance have been truly invaluable.

This project would not have been possible without the collective support of everyone mentioned above, and I remain sincerely grateful for their contributions.

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI**

**(APRIL 2025)**

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)**

**WILP Division**

**Organization:** Dell Technologies **Location:** Bangalore, Karnataka

**Duration:** 4 Months **Date of Start:** 18th January 2025

**Date of Submission:** 24th April 2025

**Title of the Project:** AI-Driven Adaptive Security and Recovery Pipeline

**ID No.:** 2023MT93089

**Name of the student:** Dabhade Pooja Bhanudas

**Name (s) and**

**Designation (s) of**

**your Supervisor and Additional Examiner:**

Rupam Kumar Kundu

Senior Software Engineer at Dell Technologies

**Name of the Faculty mentor:** Rupam Kumar Kundu

**Key Words:**

* + - * AI-Driven Security
      * Adaptive Security
      * Threat Detection
      * Anomaly Detection
      * Machine Learning in Security
      * Reinforcement Learning for Security
      * Ransomware Resilience
      * Security Automation
      * Data Integrity Protection
      * Behavioural Analysis
      * Quarantine Mechanism
      * Self-Healing Systems

**Project Areas:**

The project spans multiple research domains, including:

* Cybersecurity: Intrusion detection, ransomware resilience, and incident response.
* Artificial Intelligence: Anomaly detection, threat classification, and autonomous decision-making.
* Machine Learning: Supervised/unsupervised learning and ensemble methods.
* System Resilience: Self-healing systems and disaster recovery.
* Digital Signatures and Certificate Validation: Ensuring data integrity and secure backup storage. Blockchain: Advanced option for tamper-proof backup solutions.

# ABSTRACT

In the face of evolving cyber threats and increasing vulnerabilities, ensuring robust security and resilience for software systems has become more critical than ever. Traditional cybersecurity solutions often focus on isolated aspects such as malware detection or log analysis, leaving significant gaps in comprehensive threat management and system recovery.

This project introduces the **AI-Driven Adaptive Security and Recovery Pipeline (ASRP)**—a modular and scalable framework that integrates **monitoring, detection, quarantine, and resolution** mechanisms to provide holistic protection against a wide range of attacks, including digital intrusions, ransomware, system hardware failures, and software vulnerabilities.

The ASRP leverages **machine learning models for real-time anomaly detection, sandboxing techniques for threat containment, and AI-driven automated recovery** to minimize downtime and ensure system integrity. Key components include **log analysis using ELK Stack, AI-based behavioural analytics, digital signature validation, blockchain-enhanced security**, and **self-healing mechanisms** powered by containerization and automated rollback strategies. By combining predictive analytics with autonomous recovery capabilities, ASRP enhances resilience against both known and zero-day attacks, reducing dependency on manual intervention.

This project not only aims to **develop a working prototype** but also to **evaluate its performance against existing cybersecurity solutions**. By bridging the gaps in current security frameworks, the ASRP sets a new benchmark in **proactive threat management and intelligent system recovery**, ensuring minimal business disruption in the face of modern cyber threats.

**Signature of the Student: Signature of the Supervisor:**

** A close up of a paper

Description automatically generated**

**Name: Dabhade Pooja Bhanudas Name: Rupam Kumar Kundu**

**Place: Pune, Maharashtra Place: Bangalore, Karnataka**

**Date: 17th April 2025 Date: 17th April 2025**

# CERTIFICATE FROM THE SUPERVISOR

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI**

**CERTIFICATE**

This is to certify that the Dissertation entitled **AI-Driven Adaptive Security and Recovery Pipeline** and submitted by **Miss Pooja Dabhade** having ID No. **2023MT93089** in partial fulfilment of the requirements of Dissertation, embodies the work done by her under my supervision.

**Signature of the Supervisor:**

**A close up of a paper

Description automatically generated**

**Name: Rupam Kumar Kundu,**

**Designation: Senior Software Engineer at Dell Technologies**

**Place: Bangalore, Karnataka**

**Date: 17th April 2025**

TABLE OF CONTENTS

[ACKNOWLEDGEMENTS 3](#_Toc194493300)

[ABSTRACT 5](#_Toc194493301)

[CERTIFICATE FROM THE SUPERVISOR 7](#_Toc194493302)

[INTRODUCTION 9](#_Toc194493303)

[Scope and Limitations 9](#_Toc194493304)

[Methodology and Data Collection 10](#_Toc194493305)

[Significance and Need for This Project 10](#_Toc194493306)

[Functional block diagram of pipeline 11](#_Toc194493307)

[Modular representation of pipeline 14](#_Toc194493308)

[IDEA OVERVIEW AND WORKING OF DIFFERENT MODULES 17](#_Toc194493309)

[Stage 1: Monitoring 17](#_Toc194493310)

[Stage 2: Detection 17](#_Toc194493311)

[Stage 3: Quarantine 18](#_Toc194493312)

[Stage 4: Resolution 18](#_Toc194493313)

[LITERATURE SURVEY 19](#_Toc194493314)

[ALGORITHMS USED IN PIPELINE PHASES 25](#_Toc194493315)

[Isolation Forest algorithm: 25](#_Toc194493316)

[Random Forest Classifier: 29](#_Toc194493317)

[Reinforcement learning in Quarantine phase: 33](#_Toc194493318)

[AI integration in Resolution phase: 35](#_Toc194493319)

[CONCLUSION 36](#_Toc194493320)

[FUTURE SCOPE 37](#_Toc194493321)

[REFERENCES 38](#_Toc194493322)

[APPENDIX 39](#_Toc194493323)

[GitHub link: 39](#_Toc194493324)

[Algorithms used: 39](#_Toc194493325)

[Screenshots of code output: 41](#_Toc194493326)

[CHECKLIST OF ITEMS FOR THE FINAL DISSERTATION REPORT 45](#_Toc194493327)

# INTRODUCTION

In today's digital landscape, cybersecurity threats are growing in complexity and frequency, posing significant risks to businesses, individuals, and critical infrastructures. Traditional security solutions focus on isolated aspects such as antivirus programs, firewall protections, and log monitoring. However, these approaches often fail to provide **comprehensive protection** against sophisticated attack vectors like ransomware, digital intrusions, software vulnerabilities, and system hardware failures. The increasing reliance on cloud computing, IoT devices, and AI-driven applications further amplifies the need for **a holistic, intelligent, and adaptive security solution** that not only detects and mitigates threats but also ensures **seamless recovery with minimal downtime**.

This project, titled **"AI-Driven Adaptive Security and Recovery Pipeline (ASRP),"** aims to bridge these security gaps by developing a **modular, scalable, and AI-powered security framework**. The **ASRP integrates four core functionalities—Monitoring, Detection, Quarantine, and Resolution—to provide end-to-end threat management and system recovery.**

* **Monitoring:** Utilizes AI-driven anomaly detection to analyse system logs, network traffic, and hardware performance.
* **Detection:** Implements a hybrid approach combining **signature-based** and **behavioural analytics** for real-time threat classification and risk assessment.
* **Quarantine:** Introduces **sandboxing mechanisms** and **digital signature verification** to isolate compromised components and maintain data integrity.
* **Resolution:** Employs **automated rollback mechanisms, self-healing techniques, and AI-driven decision-making** to restore system functionality with minimal human intervention.

## Scope and Limitations

The **scope** of this project extends to cybersecurity domains including **intrusion detection, automated threat mitigation, and intelligent recovery systems.** By leveraging **machine learning models, blockchain-based verification, and containerization techniques**, ASRP enhances system resilience against **both known and zero-day threats**.

However, certain **limitations** must be acknowledged. The project primarily focuses on enterprise environments and **may require customization for industry-specific implementations**.

Additionally, **AI-based threat detection models require continuous training and updates** to remain effective against emerging cyber threats. The integration of **advanced blockchain mechanisms for tamper-proof security** also presents scalability challenges that need further exploration.

## Methodology and Data Collection

The ASRP is developed using a **combination of AI-driven security tools, open-source cybersecurity frameworks, and cloud-based infrastructure**. The methodology includes:

1. **Literature Review** – Researching existing cybersecurity solutions and identifying gaps in traditional security models.
2. **Data Collection** – Using system logs, network traffic, and attack datasets to train machine learning models.
3. **Model Development** – Implementing **AI-based detection mechanisms, sandboxing strategies, and automated rollback systems**.
4. **Prototype Development & Testing** – Deploying the pipeline in a **simulated environment** and evaluating its performance against real-world cyber threats.

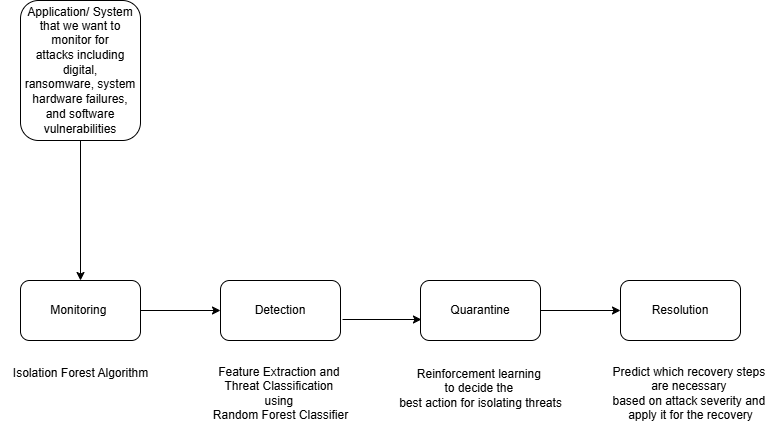
## Significance and Need for This Project

With cyberattacks becoming increasingly sophisticated, relying on **reactive security measures is no longer sufficient**. The **AI-Driven Adaptive Security and Recovery Pipeline (ASRP) introduces a proactive and autonomous defence mechanism**, reducing the need for manual intervention in threat mitigation and system recovery.

By integrating **advanced AI-driven analytics, blockchain-backed data verification, and automated rollback mechanisms,** ASRP sets a new benchmark for intelligent cybersecurity solutions.

This project contributes to the evolving field of **cybersecurity research and enterprise resilience**, making it a valuable framework for organizations seeking **enhanced security posture, reduced downtime, and effective risk mitigation strategies**.

## Functional block diagram of pipeline



The **Functional Block Diagram** illustrates the **AI-Driven Adaptive Security and Recovery Pipeline (ASRP)**, which consists of four key phases: **Monitoring, Detection, Quarantine, and Resolution**. Each phase is integrated with AI and machine learning techniques to enhance system security, detect anomalies, and automate recovery processes.

**1. Monitoring Phase**

Objective:

Continuously observe the system to identify anomalies or suspicious activities.

* Utilizes the Isolation Forest Algorithm, an unsupervised machine learning technique, to detect outliers in system logs, network traffic, and hardware performance.
* Captures deviations from normal behaviour and flags potential security threats.
* Outputs flagged instances for further analysis in the Detection Phase.

Example:

If a sudden surge in CPU usage or unauthorized file modifications is detected, the system raises an alert for deeper investigation.

**2. Detection Phase**

Objective:

Perform detailed analysis and classify threats based on severity and type.

* Implements Feature Extraction and Threat Classification using Random Forest Classifier, which is a supervised learning algorithm.
* Analyses flagged anomalies from the Monitoring Phase and determines whether they are genuine threats or false positives.
* Classifies threats into categories such as malware, ransomware, unauthorized access, or system failure.
* Prioritizes high-risk threats and forwards them to the Quarantine Phase for containment.

Example:

If an unauthorized executable is running in the system, the classifier determines whether it is a benign or malicious process.

**3. Quarantine Phase**

Objective:

Isolate and contain detected threats to prevent system-wide damage.

* Uses Reinforcement Learning to decide the best isolation mechanism based on historical attack data.
* Implements sandboxing techniques to restrict the movement of malicious binaries or unauthorized processes.
* Prevents infected or compromised components from interacting with the rest of the system.

Example:

If ransomware activity is detected, the system automatically isolates the infected files to stop encryption and prevent data loss.

**4. Resolution Phase**

Objective:

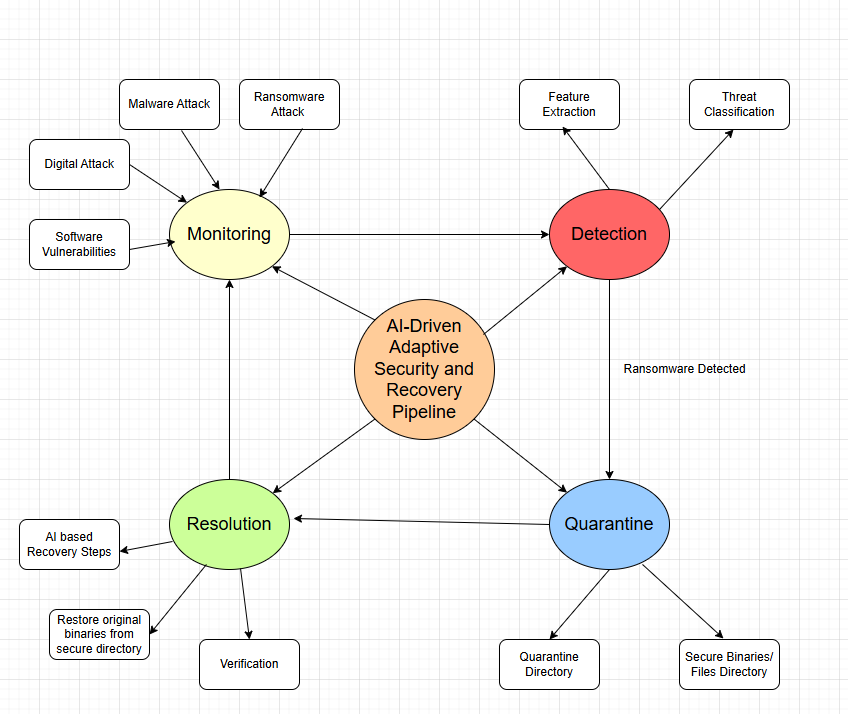
Recover affected components and restore system integrity.

* AI-based predictive recovery models determine the necessary remediation actions based on attack severity.
* Automatically applies the best recovery method, such as rolling back to a secure snapshot, replacing compromised files, or restoring from a backup.
* Ensures minimal downtime and business continuity.

Example:

If a system library is compromised, the system replaces it with a verified version from a secure repository and restores normal functionality.

## Modular representation of pipeline



The AI-Driven Adaptive Security and Recovery Pipeline (ASRP) is designed to provide a structured and automated approach to cybersecurity by integrating multiple defence mechanisms. The diagram represents the four core modules: Monitoring, Detection, Quarantine, and Resolution, each having specific submodules that contribute to the overall security and recovery process.

**1. Monitoring Module**

The Monitoring module is responsible for continuously tracking system activities to detect potential threats. It identifies various types of attacks, including:

Malware Attacks – Harmful software designed to disrupt, damage, or gain unauthorized access.

Ransomware Attacks – A form of malware that encrypts files and demands payment for decryption.

Digital Attacks – General cyber threats, such as unauthorized access or data breaches.

Software Vulnerabilities – Weaknesses in software that can be exploited by attackers.

This module forms the first line of defence by detecting suspicious activities and forwarding them to the Detection module for further analysis.

**2. Detection Module**

Once an anomaly is detected, the Detection module performs further analysis to classify and verify the threat. It consists of:

Feature Extraction – Identifies key characteristics from system logs and network traffic to distinguish between normal and abnormal activities.

Threat Classification – Uses machine learning techniques (such as Random Forest Classifier) to categorize the detected anomalies as genuine threats or false positives.

If an attack is confirmed, it triggers the Quarantine module for immediate containment.

**3. Quarantine Module**

The Quarantine module isolates and contains detected threats to prevent further spread. It includes:

Quarantine Directory – Stores affected files and binaries in a secure location, preventing further execution.

Secure Binaries/Files Directory – Maintains a repository of clean and verified files that can be used for replacement during recovery.

This module ensures that the compromised elements are properly contained before initiating the Resolution phase.

**4. Resolution Module**

The Resolution module is responsible for restoring system integrity and ensuring that affected components are replaced with secure versions. It includes:

AI-Based Recovery Steps – Uses intelligent decision-making to determine the best approach for restoring affected components.

Restore Original Binaries from Secure Directory – Retrieves clean, verified files to replace corrupted or quarantined binaries.

Verification – Ensures that the restored components are functioning correctly and the system is secure before resuming normal operations.

# IDEA OVERVIEW AND WORKING OF DIFFERENT MODULES

## **Stage 1: Monitoring**

**Objective:**

Simulate system monitoring for suspicious activity (e.g., checking file changes in a directory where application files are stored).

**Implementation:**

- Monitor a directory of any application which we want to monitor for any unauthorized changes (e.g., creation of an encrypted file).

- Log events like file modifications, deletions, or new processes related to application.

**AI Integration:**

- Use an anomaly detection model (e.g., using Long Short-Term Memory (LSTM) networks, Isolation Forests, or Autoencoders) to identify unusual patterns in file activities or system behaviours that deviate from normal baselines.

- Example: Train an AI model to recognize typical file access patterns in the Notepad directory. Flag abnormal activities such as rapid file encryption (ransomware-like behaviour).

## **Stage 2: Detection**

**Objective:**

Detect suspicious patterns indicating an attack.

**Implementation:**

- Use a lightweight anomaly detection algorithm (e.g., comparing hash values of application binaries).

- Trigger alerts when unauthorized modifications to files or binaries are detected.

**AI Integration:**

- Implement machine learning models (e.g., Random Forests, Gradient Boosting, or SVM) for classifying activities as malicious or benign based on features extracted from:

* File metadata (size, type, etc.).
* System events (process creation, network connections).
* Behavioural patterns (frequency of file modifications).

- Example: Use a pre-trained malware classification model to analyse and score file or process activities.

## **Stage 3: Quarantine**

**Objective:**

Isolate and block malicious files or processes.

**Implementation:**

- Move suspicious files to a quarantine directory.

- Replace compromised binaries with secure ones from a pre-verified repository.

**AI Integration:**

- Apply reinforcement learning to decide the best action for isolating threats (e.g., blocking processes vs. quarantining files).

- Example: The system could dynamically adjust quarantine thresholds based on live threat assessments using AI.

## **Stage 4: Resolution**

**Objective:**

Automatically recover from the detected attack.

**Implementation:**

- Restore original files and binaries from the verified repository.

- Restart the application and verify the integrity of the restored components.

**AI Integration:**

- Employ AI models to:

• Predict which recovery steps are necessary based on attack severity.

• Validate the restored application by simulating expected behaviour and ensuring its integrity.

- Example: Use a model trained on past recovery logs to recommend whether to simply replace the file or take additional steps like restarting related services or notifying admins.

# LITERATURE SURVEY

There are some already existing systems which are part of the cybersecurity ecosystem, designed to protect organizations against evolving threats. Each focus on specific aspects of security, ranging from detection and prevention to response and remediation.

**1. IBM QRadar:**

**Category:**

Security Information and Event Management (SIEM)

**Purpose:**

Log analysis, threat intelligence, and correlation of security data.

**Key Functions:**

- Centralized Log Management: Collects logs from various sources (firewalls, servers, applications) and correlates them.

- Threat Detection: Uses rule-based and AI/ML-driven analytics to detect anomalies or potential threats.

- Incident Response: Provides insights for security teams to respond to incidents faster.

- Compliance Management: Ensures compliance with regulations like GDPR, HIPAA, or PCI-DSS.

**Usage Scenario:**

Large enterprises with diverse IT infrastructures needing centralized security monitoring and compliance reporting.

**2. CrowdStrike Falcon:**

**Category:**

Endpoint Detection and Response (EDR) and Managed Detection and Response (MDR)

**Purpose:**

Protects endpoints (desktops, laptops, servers) from threats like malware, ransomware, and zero-day exploits.

**Key Functions:**

- Endpoint Protection: Real-time monitoring and detection of malicious activities on endpoints.

- Threat Hunting: Uses AI to detect sophisticated threats and provides tools for human threat hunters.

- Incident Response: Identifies and isolates compromised endpoints to prevent lateral movement.

- Cloud-Based: Operates through lightweight agents that report to a cloud platform for analysis.

**Usage Scenario:**

Organizations seeking robust, real-time endpoint security with AI-driven insights and fast remediation capabilities.

**3. SentinelOne:**

**Category:**

Endpoint Protection Platform (EPP) with AI and EDR capabilities

**Purpose:**

Autonomous detection, prevention, and remediation for endpoint attacks.

**Key Functions:**

- AI-Powered Threat Detection: Uses ML models to detect and block known and unknown threats in real time.

- Behavioural Analysis: Tracks application and system behaviours to identify anomalies and predict attacks.

- Automated Remediation: Automatically isolates and rolls back systems to a pre-attack state.

- Ransomware Mitigation: Protects against ransomware by detecting encryption patterns and terminating malicious processes.

**Usage Scenario:**

Businesses needing an all-in-one, autonomous solution for endpoint protection and fast recovery.

**comparison table between different existing solutions:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Feature** |  |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  | | **IBM QRadar (SIEM)** | **CrowdStrike Falcon (EDR/MDR)** | **SentinelOne (EPP/EDR)** | **Proposed Solution (ASRP)** |
| **Focus** | Log analysis & threat detection | Endpoint security & threat hunting | Autonomous endpoint protection | Full-cycle monitoring, detection, isolation, and recovery |
| **Detection Method** | Log correlation & AI-based analysis | AI-driven real-time monitoring | AI-powered behaviour tracking | Isolation Forest & Random Forest for detection |
| **Threat Containment** | Requires manual response | Isolates infected endpoints | Automatically blocks threats | AI-driven quarantine of threats |
| **Automated Recovery** | No automated rollback | |  | | --- | | Limited recovery features |  |  | | --- | |  |  |  | | --- | |  | | Self-healing rollback | AI-based recovery with secure file replacement |
| **Ransomware Protection** | |  | | --- | | Detects ransomware via logs |  |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  | | Identifies & isolates infected endpoints | Blocks ransomware encryption patterns | Detects, isolates, and recovers from ransomware attacks |
| **Human Intervention** | Requires security analysts | AI-assisted, but needs monitoring | Minimal human input | Fully automated response system |

**To differentiate my project from existing solutions like IBM QRadar, CrowdStrike Falcon, and SentinelOne, I am focusing on filling the gaps and introducing novel features. This project stands out by providing a comprehensive, adaptive, and unified solution that combines the strengths of existing tools while addressing their limitations. Focus on automation, predictive capabilities, and hardware-software integration for a truly innovative approach. Here's a list of potential enhancements my project can offer:**

* **AI-Powered Autonomous Quarantine and Recovery:**

Current Limitation:

- Most tools rely on manual intervention for certain remediation steps (e.g., deciding what to isolate or rollback).

Enhancement:

- Autonomous Quarantine: Automatically isolate compromised binaries or system components using predictive analytics.

- Intelligent Recovery: Implementing self-healing mechanisms that dynamically replace infected binaries from a secure, blockchain-backed repository.

* **Cross-Platform Integration:**

Current Limitation:

- Existing tools are often tied to specific environments (e.g., IBM QRadar for centralized logs, SentinelOne for endpoints).

Enhancement:

- To Build a system that seamlessly integrates with multiple platforms and environments (on-premises, cloud, hybrid) and coordinates across all levels of infrastructure (databases, endpoints, applications, and servers).

* **Zero-Day Attack Prediction and Prevention:**

Current Limitation:

- Many tools focus on detecting attacks after they happen or blocking known threats.

Enhancement:

- Introduce predictive models for zero-day vulnerabilities by leveraging large-scale data and transfer learning.

- Use adversarial learning to simulate potential zero-day exploits and prepare proactive defences.

* **End-to-End Visibility with Multi-Layered Dashboards:**

Current Limitation:

- Dashboards in existing tools often focus on specific aspects (logs, endpoints).

Enhancement:

- Design multi-layered dashboards providing real-time visibility across:

- Hardware health.

- Software logs.

- Network activity.

- Threat status and resolution progress.

* **AI-Powered Threat Attribution:**

Current Limitation:

- Tools focus on stopping threats but don't always identify their origins.

Enhancement:

- Implement AI models for threat attribution to track and identify the source of the attack (e.g., insider threats, supply chain attacks).

* **Blockchain for Integrity and Auditability:**

Current Limitation:

- Lack of tamper-proof mechanisms to verify the integrity of quarantined data or logs.

Enhancement:

- Use blockchain for storing:

- System state snapshots.

- Integrity-verified backups of binaries.

- Immutable audit trails for post-incident investigations.

* **AI-Guided Recovery Prioritization:**

Current Limitation:

- Existing systems treat all threats as equal in urgency.

Enhancement:

- Develop an AI-guided prioritization engine that assesses the severity and business impact of threats and recommends a response order.

* **Modular and Extensible Design:**

Current Limitation:

- Many existing tools are rigid and don't allow seamless expansion for new attack types.

Enhancement:

- Designing a system as a modular framework where:

- New AI models or attack-specific detection modules can be plugged in.

- Organizations can customize based on their specific needs.

* **Self-Adaptive Threat Models:**

Current Limitation:

- Static models may fail to adapt to new threat patterns.

Enhancement:

- Incorporating self-learning AI models that continuously evolve by incorporating new threat data into their training.

# ALGORITHMS USED IN PIPELINE PHASES

## **Isolation Forest algorithm:**

The Isolation Forest algorithm is a machine learning method specifically designed to detect anomalies (unusual patterns) in a dataset. It works by isolating data points that behave differently from the majority

* The Basic Idea:

- Imagine your data points are trees in a forest.

- To isolate a tree (data point), you chop through branches (features of the data) until the tree is cut off from the forest.

- Trees that are very different from the others (anomalies) can be isolated quickly with fewer cuts.

- Trees that are similar to the others (normal points) need more cuts because they’re deeply connected to the rest of the forest.

* How It Works:

1. Random Splitting:

- The algorithm builds a forest of "isolation trees" by splitting the data repeatedly at random.

- For each split, it picks a feature (e.g., file size, modification frequency) and a random value to divide the data.

2. Isolation Depth:

- The more splits (cuts) it takes to isolate a data point, the more "normal" it is.

- If a point is isolated in just a few splits, it’s likely an anomaly.

3. Anomaly Score:

- Each point gets a score based on how quickly it was isolated across all the trees.

- A high score means the point is likely an anomaly.

Why It’s Great:

- Efficient: It’s faster than other anomaly detection methods because it doesn’t need to calculate distances or probabilities for every data point.

- Scalable: Works well even with large datasets.

- No Assumptions: It doesn’t assume the data follows a specific distribution (like normal distribution).

* Real-Life Example:

- You monitor file sizes in a directory.

- Most files are between 90 KB and 110 KB.

- Suddenly, a file of 5000 KB appears.

- The Isolation Forest quickly isolates this file because it’s very different from the others, flagging it as an anomaly.

* Key Parameters:

- Number of Trees: More trees give better results but take longer to compute.

- Contamination: This is the proportion of data you expect to be anomalies (e.g., 5%).

In simple terms, the Isolation Forest is like a detective looking for suspicious behaviour in a crowd. It focuses on how quickly it can "single out" someone acting differently from the rest. If it’s too easy to isolate, it’s flagged as suspicious!

* Why Choose Isolation Forest for Monitoring Phase?

- Efficiency:

Isolation Forest is computationally efficient because it isolates anomalies using random splits rather than calculating distances or densities.

Ideal for real-time monitoring in systems with frequent updates.

- Scalability:

It works well with large datasets, which is critical for monitoring logs, file systems, or networks.

- Simplicity:

Requires fewer parameters to tune compared to methods like DBSCAN or Autoencoders.

- No Distribution Assumption:

Unlike PCA or One-Class SVM, Isolation Forest does not assume that the data follows a specific distribution, making it versatile for different datasets.

- Robust to High Dimensions:

Handles datasets with many features (e.g., file size, modification time, frequency) without suffering performance degradation.

- Interpretability:

Anomaly scores are intuitive and easy to understand. For example, a high anomaly score directly means the instance is more likely to be anomalous.

* Key Trade-offs:

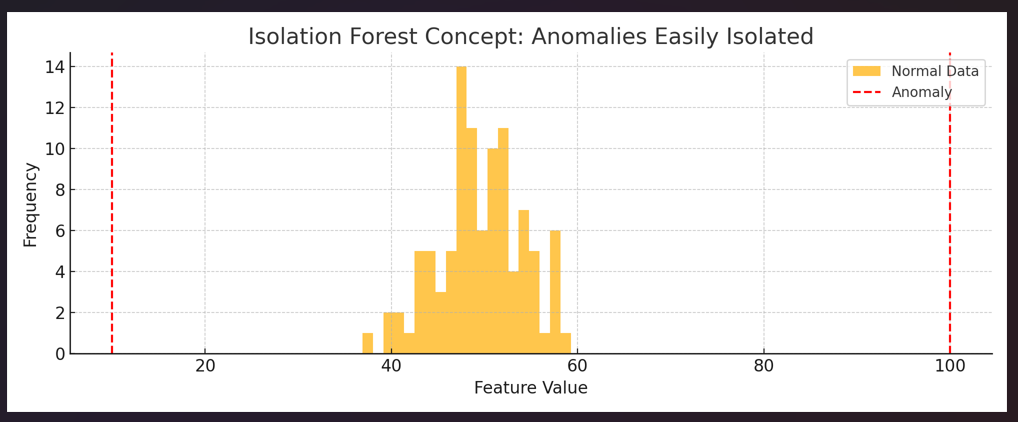
- Why Not Use Neural Networks (Autoencoders)?

Autoencoders are powerful but require large amounts of training data and computational power, which might not be feasible for lightweight systems or quick prototyping.

- Why Not Use Density-Based Methods (DBSCAN, LOF)?

These methods struggle with high-dimensional data and require precise tuning of parameters, which can complicate implementation in dynamic monitoring systems.

* Isolation Forest Concept Graph:



The graph illustrates how the **Isolation Forest algorithm** identifies anomalies by isolating unusual data points from normal ones.

* Most of the data (normal file sizes) are clustered around the centre of the graph.
* The two red lines at the edges (10 and 100) represent **anomalous file sizes**.
* Since these points are far from the majority, the algorithm can **easily separate them** with fewer steps, marking them as anomalies.

This visual shows that **outliers stand apart from regular patterns**, and Isolation Forest quickly detects them based on how easy they are to isolate.

* Anomaly Detection Algorithm Comparison:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Algorithm** |  |  | | --- | |  |  |  | | --- | |  | | |  | | --- | | **Strengths** | | **Limitations** | **Why/Why Not for This Project** |
| Isolation Forest | - Fast & scalable - Works with high-dimensional data - No distribution assumptions | - May not handle complex non-linear patterns | ✅ **Chosen** for its speed, scalability, and effectiveness in real-time anomaly detection. Ideal for large log data and file monitoring. |
| One-Class SVM | - Effective in high-dimensional space - Good for small datasets | - Not scalable - Sensitive to outliers in training data | ❌ Slower and less scalable. Not suitable for large system logs. |
| DBSCAN | - Handles arbitrary cluster shapes - non-parametric | - Struggles with high dimensions - Parameter tuning needed | ❌ Inefficient with high-dimensional security data. |
| Autoencoders | - Powerful for complex data - Learns non-linear patterns | - Requires a lot of data - High computational cost | ❌ Good for deep learning tasks, but overkill for lightweight monitoring. |
| LOF (Local Outlier Factor) | - Detects local anomalies - No distribution assumptions | - High computation - Sensitive to parameter selection | ❌ Slower for real-time use and not as efficient on large logs. |
| PCA | - Simple - Useful for dimensionality reduction | - Assumes linearity - Sensitive to noise | ❌ Too simplistic for complex attack patterns in security data. |

* Conclusion:

Isolation Forest is a fast, scalable, and robust algorithm that fits the requirements of the Monitoring Phase in the pipeline. It ensures the system can efficiently and effectively flag anomalies in real-time without excessive computational overhead.

## **Random Forest Classifier:**

* How It Works:

Training:

The model learns patterns from the training dataset, such as how large file sizes or recent modifications correlate with specific threats.

Prediction:

For unseen data, the model predicts the likelihood of each threat type based on learned patterns.

Evaluation:

The classification report shows how well the model performs on unseen data.

* Why Use AI Here?

- AI-powered models can handle complex patterns in data that traditional rules-based systems might miss.

- Over time, the model improves as it learns from new data, adapting to emerging threats like new types of malware or phishing attacks.

* Input to the Detection Stage:

- size: The file's size in bytes.

- modification\_time: The timestamp when the file was last modified.

- creation\_time: The timestamp when the file was created.

- anomaly: Indicates the anomaly score (-1 means anomalous as flagged by the Isolation Forest algorithm).

* How the Detection Stage Works:

Feature Extraction:

- The Detection Stage uses the size, modification\_time, and creation\_time features as input.

- The anomaly score is not used in this stage because the Detection Stage focuses on identifying the type of threat.

* Threat Classification:

- The Random Forest Classifier (or another ML model) analyses the input data.

- Based on the patterns learned during training, the model assigns the file to a specific threat category (e.g., benign, ransomware, phishing).

For instance:

- Large file size + recent modification = ransomware.

- Sudden appearance of a new file + no recent modification = phishing.

* Output of the Detection Stage:

- Predicted Threat:

The Detection Stage classifies the file as Ransomware, meaning it suspects that the anomaly is caused by a ransomware attack.

- Confidence:

This is the model's certainty in its prediction.

For example:

85% means the model is confident that the anomaly represents ransomware.

* How to Interpret This Process:

- In the Monitoring Stage, the anomaly was flagged based on statistical irregularities (e.g., unusual size or timestamps).

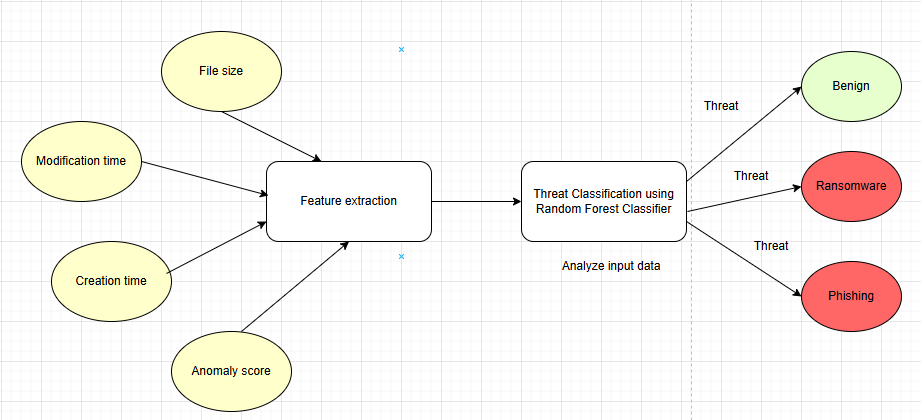
- In the Detection Stage, the anomaly is classified into a specific threat category using a trained model.

- If the model's confidence is low (e.g., <50%), the output should be treated with caution, and further analysis (e.g., manual review) might be required.

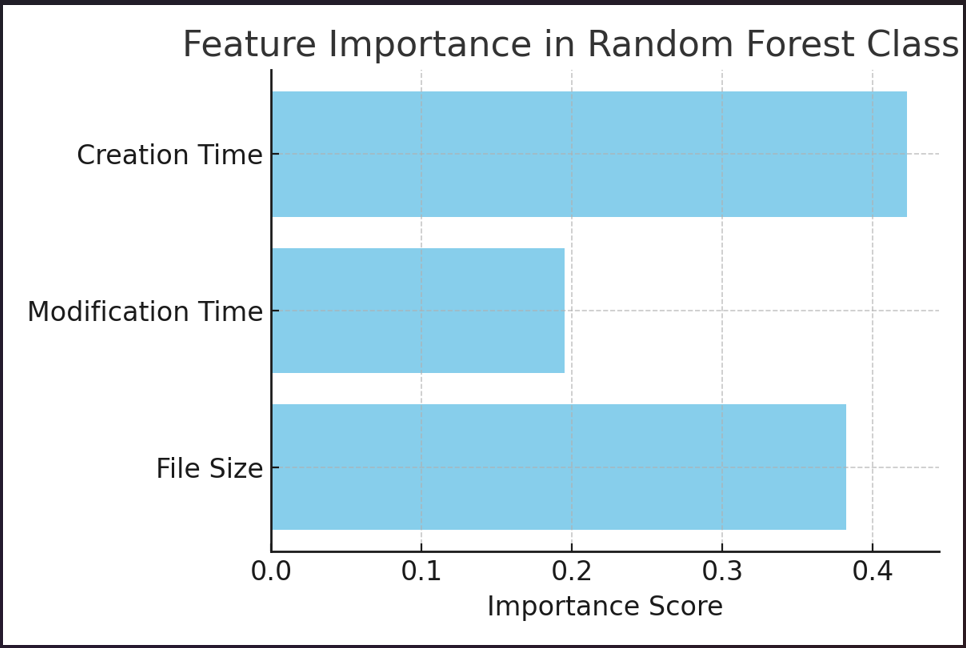
- The stage adds context to the anomaly, helping you decide the next action:

- If ransomware is detected, the pipeline moves to the Quarantine Stage.

- If the anomaly is benign, no action is taken.



* Feature Importance Graph – Random Forest:



- This graph shows how much each feature (File Size, Modification Time, Creation Time) contributes to the threat classification decision.

- Features with higher importance scores play a larger role in determining whether a file is benign or malicious.

* Sample Decision Tree from Random Forest:

A diagram of a computer algorithm

AI-generated content may be incorrect.

- This diagram represents one of the decision trees from the Random Forest.

- It visualizes how the model makes branching decisions based on feature thresholds to classify a file (e.g., as Ransomware or Phishing).

## **Reinforcement learning in Quarantine phase:**

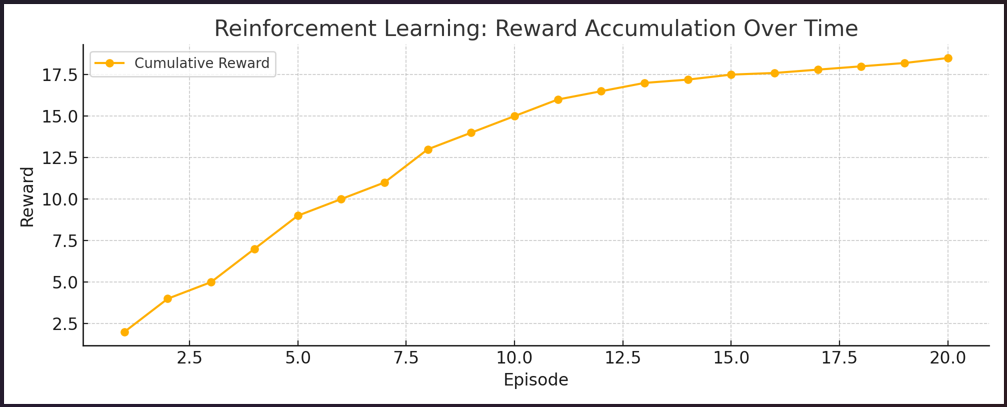
The **Quarantine Stage** is responsible for isolating and neutralizing malicious files or processes to prevent further damage to the system. While basic quarantine mechanisms involve moving suspicious files to a safe directory or replacing them with clean versions, **AI enhances this stage by making the process smarter and more adaptive.**

**1. Reinforcement Learning (RL) for Dynamic Decision-Making**

* **Reinforcement Learning** enables the system to **learn optimal quarantine actions over time**.
* The AI agent observes various threat scenarios and receives feedback based on the effectiveness of actions (e.g., isolating a file vs. killing a process).
* Over time, it learns the best strategy for different threat types, improving accuracy and reducing false positives.

**Example:**

* If isolating a suspicious file successfully prevents further infection, the model rewards that decision.
* If terminating a process proves more effective for stopping ransomware encryption, that strategy is reinforced.
* **RL Reward Accumulation Graph:**



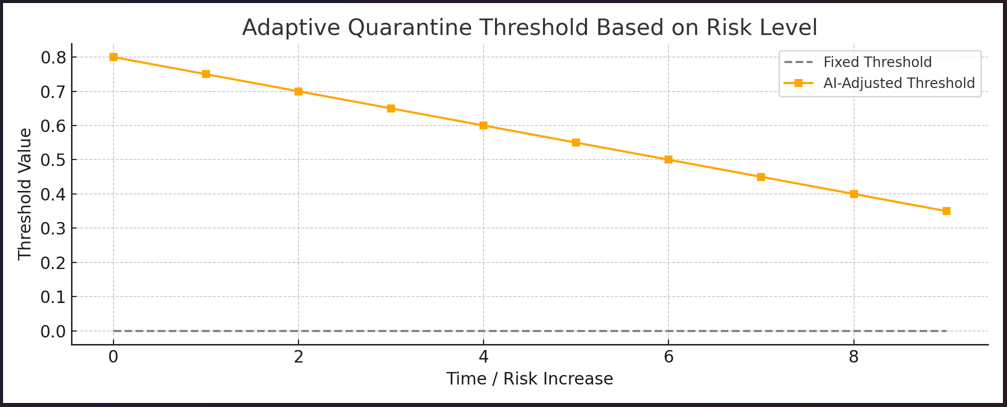
* + Shows how a Reinforcement Learning (RL) agent learns over time by accumulating rewards.
  + Higher rewards reflect better decision-making (e.g., effectively isolating threats).

**2. Adaptive Quarantine Thresholds**

* Instead of using fixed rules, AI dynamically **adjusts quarantine sensitivity** based on current threat levels or environmental changes.
* During a spike in anomalous activity (e.g., during a ransomware outbreak), the system can automatically lower the threshold, becoming **more aggressive in isolating files.**

**Example:**

* During normal system behaviour: only highly suspicious files are quarantined.
* During high-risk periods: even moderately suspicious activity can trigger isolation, based on **real-time AI-driven risk scoring**.
* **Adaptive Quarantine Threshold Graph:**



* + Illustrates how AI can dynamically lower the quarantine threshold based on increasing risk.
  + Unlike a fixed threshold, the adaptive one responds to real-time threat levels, becoming more aggressive as threats rise.

## **AI integration in Resolution phase:**

The **Resolution Stage** is responsible for restoring the system to a secure and functional state after a threat has been detected and contained. While traditional recovery systems rely on fixed scripts or manual intervention, **AI integration enables intelligent, adaptive, and context-aware recovery decisions**—ensuring faster, safer, and more reliable restoration.

**1. AI-Based Recovery Decision-Making**

* AI models analyse the **severity and type of attack** (ransomware, software corruption, unauthorized access, etc.) and recommend the most suitable recovery steps.
* The system can **differentiate between minor disruptions (e.g., corrupted config file)** and **major breaches (e.g., infected system binaries)** and decide whether:
  + To **replace the file only**,
  + Restart the **entire application**, or
  + Notify **system administrators**.

**Example:**

* If a single DLL is flagged and isolated, the AI may decide that replacing it and restarting the service is sufficient.
* If multiple components were affected within seconds, AI may trigger a full rollback and raise an alert.

**2. Simulation & Post-Recovery Validation**

* After restoring files from a **pre-verified repository**, AI models simulate the **expected behaviour of the application** (e.g., process patterns, resource usage).
* If discrepancies are observed (e.g., unexpected memory spikes or unauthorized file access), AI **flags the restoration as potentially incomplete or risky**, prompting further actions.

**Example:**

* The model knows how Notepad++ behaves normally. If after recovery, the process still attempts unusual file access, AI suspects deeper compromise and recommends isolating again or alerting the admin.

**3. Learning from Recovery History**

* AI can be trained using **past recovery logs** (e.g., attack types, actions taken, recovery success/failure).
* Over time, it **learns which recovery strategies are most effective** under specific conditions and improves its recommendations accordingly.

**Example:**

* If past logs show that simply restoring files was not enough for ransomware attacks, AI will recommend **rebooting the host service and running a post-recovery scan** in future cases.

# CONCLUSION

This dissertation presents an **AI-Driven Adaptive Security and Recovery Pipeline (ASRP)**, a comprehensive solution designed to **mitigate cybersecurity threats, enhance system resilience, and automate recovery mechanisms**.

As cyberattacks continue to evolve in complexity, traditional security measures often fall short in **detecting, isolating, and recovering from threats in a timely manner**. This project bridges those gaps by leveraging **artificial intelligence, machine learning, and automation** to build an intelligent **self-healing security framework** capable of **proactively monitoring, detecting, quarantining, and resolving** security threats with minimal human intervention.

Throughout the development of ASRP, various **machine learning algorithms** were employed to enhance different stages of the pipeline:

* **Monitoring:** The **Isolation Forest algorithm** was used for **anomaly detection** in system logs, network traffic, and hardware metrics to identify deviations from normal behaviour, helping detect early signs of security threats.
* **Detection:** **Feature extraction and threat classification** were performed using the **Random Forest Classifier**, which enabled accurate identification of malicious activities based on system behaviour and attack patterns.
* **Quarantine:** Digital signatures, hash verification, and **sandboxing mechanisms** were integrated to isolate and analyse potentially malicious files and processes, preventing further system compromise.
* **Resolution:** **Reinforcement learning-based decision models** were explored to automate incident response strategies, while containerization tools such as **Docker and Kubernetes** enabled rapid restoration of secure system states.

# FUTURE SCOPE

1. **Scalability to Enterprise Systems** – Expand the framework to protect large-scale enterprise applications, cloud platforms, and critical infrastructure.
2. **Advanced AI & Machine Learning Integration** – Implement deep learning models for improved threat detection, anomaly analysis, and autonomous decision-making.
3. **Enhanced Quarantine & Recovery Mechanisms** – Utilize blockchain for data integrity, automate rollback processes, and strengthen zero-trust security models.
4. **Real-Time Threat Intelligence & Adaptive Security** – Incorporate real-world attack simulations, security event monitoring, and predictive analytics for proactive defence.
5. **Industry Compliance & Customization** – Ensure alignment with global cybersecurity standards (NIST, GDPR, ISO 27001) and adapt the system for healthcare, finance, and industrial applications.

# REFERENCES

The following are referred journals from the preliminary literature review.

1. *A. Manoharan and M. Sarker, ‘‘Revolutionizing cybersecurity: Unleashing the power of artificial intelligence and ML for next-generation threat detection,’’ Tech. Rep., 2023, doi: 10.56726/IRJMETS32644.*
2. *M. Asmar and A. Tuqan, ‘‘Integrating machine learning for sustaining cybersecurity in digital banks,’’ Heliyon, vol. 10, no. 17, Sep. 2024, Art. no. e37571.*
3. *I. Kara and M. Aydos, "Cyber fraud: Detection and analysis of the crypto-ransomware", Proc. 11th IEEE Annu. Ubiquitous Comput. Electron. Mobile Commun. Conf. (UEMCON), pp. 764-769, Oct. 2020.*
4. *I. Kara and M. Aydos, "The rise of ransomware: Forensic analysis for windows based ransomware attacks", Exp. Syst. Appl., vol. 190, Mar. 2022.*
5. *A. M. Maigida, S. M. Abdulhamid, M. Olalere, J. K. Alhassan, H. Chiroma and E. G. Dada, "Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms", J. Reliable Intell. Environ., vol. 5, no. 2, pp. 67-89, Jul. 2019.*

# APPENDIX

## **GitHub link:**

<https://github.com/poojaDabhade23/AI-driven-Adaptive-Security-and-Recovery-Pipeline/tree/main/Development-Work>

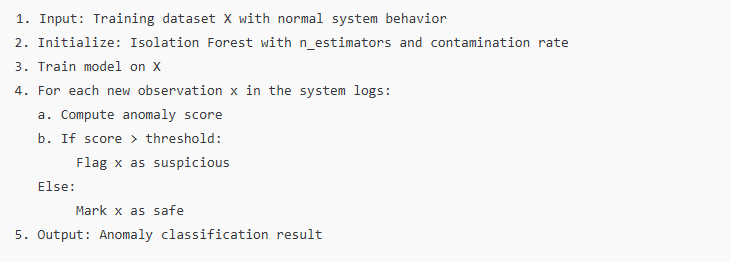
## **Algorithms used:**

**1. Isolation Forest for Anomaly Detection (Monitoring Phase)**

**Algorithm Name**: Isolation Forest  
**Purpose**: Used for detecting anomalies in system logs, network traffic, and hardware performance during the monitoring phase.  
**Steps**:

1. Train the Isolation Forest model on normal system behaviour.
2. Compute anomaly scores based on data isolation depth.
3. Flag instances with high anomaly scores as potential security threats.
4. Forward flagged instances to the detection module for further analysis.

**Pseudocode –**

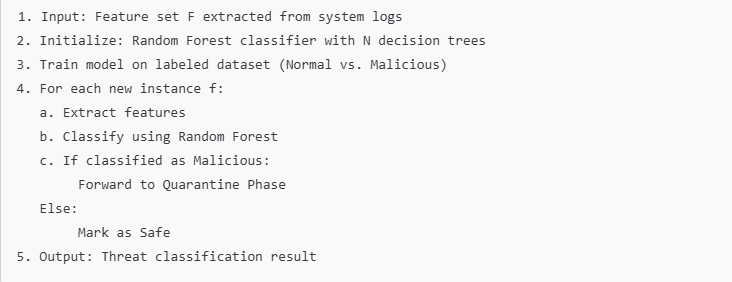


**2. Random Forest for Threat Classification (Detection Phase)**

**Algorithm Name**: Random Forest Classifier  
**Purpose**: Used for feature extraction and classifying threats based on system behaviours in the detection phase.  
**Steps**:

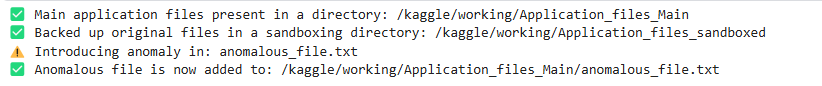
1. Extract features from system logs and network activity.
2. Train Random Forest model on labelled attack and normal datasets.
3. Predict attack types and classify threats.
4. Forward high-risk threats to quarantine or resolution phases.

**Pseudocode –**

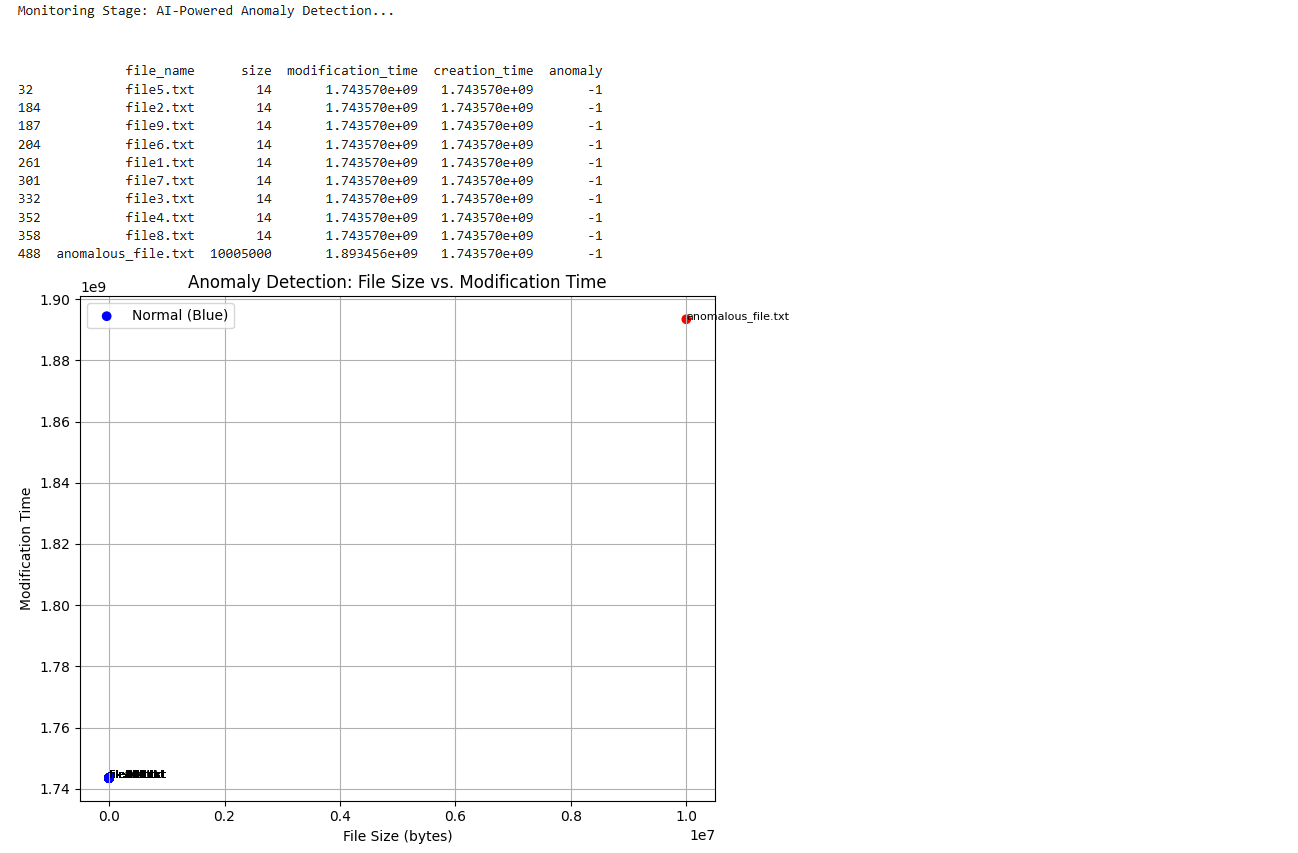


## **Screenshots of code output:**

1. Anomaly injection phase –



1. Monitoring phase –



1. Detection phase –

A screen shot of a computer screen

AI-generated content may be incorrect.

1. Quarantine phase –

A graph with red bars

AI-generated content may be incorrect.

A screenshot of a computer

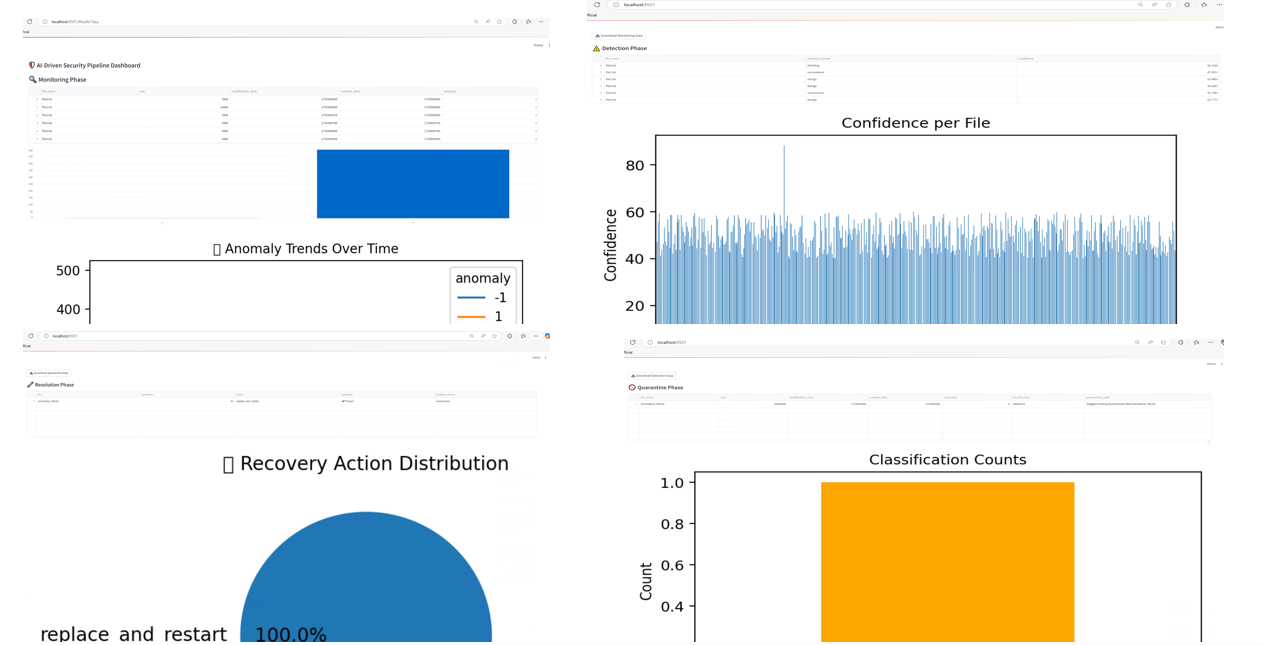
AI-generated content may be incorrect.

1. Resolution phase –

A screenshot of a computer error

AI-generated content may be incorrect.

1. Dashboarding phase –



# CHECKLIST OF ITEMS FOR THE FINAL DISSERTATION REPORT

|  |  |  |
| --- | --- | --- |
|  | Is the final report neatly formatted with all the elements required for a technical Report? | **Yes** |
|  | Is the Cover page in proper format as given in Annexure A? | **Yes** |
|  | Is the Title page (Inner cover page) in proper format? | **Yes** |
|  | (a) Is the Certificate from the Supervisor in proper format?  (b) Has it been signed by the Supervisor? | **Yes**  **Yes** |
|  | Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly? | **Yes**  **Yes** |
|  | Is the title of your report appropriate? | **Yes** |
|  | Does the Report contain a summary of the literature survey? | **Yes** |
|  | Does the Table of Contents include page numbers?   1. Are the Pages numbered properly? 2. Are the Figures numbered properly? 3. Are the Tables numbered properly? 4. Are the Captions for the Figures and Tables proper? 5. Are the Appendices numbered properly? Are their titles appropriate | **Yes**  **Yes**  **Yes**  **Yes**  **Yes**  **Yes** |
|  | Is the conclusion of the Report based on discussion of the work? | **Yes** |
|  | Are References given at the end of the Report? | **Yes** |
|  | Is the report format and content according to the guidelines? | **Yes** |

**Declaration by Student:**

I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.

**Place: Pune                          Signature of the Student:**

****

**Date:  17th April 2025                         Name: Dabhade Pooja Bhanudas**

**ID No.: 2023MT93089**