iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** III **Month of publication:** March 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.79086

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue III Mar 2026- Available at www.ijraset.com*

# An Intelligent AI Ops Framework for Automated Failure Detection and Recovery in Digital Workflows

M.Yazhini[1], S.Swathi[2], A.Akileshwari[3]
[1, 2]*UG Scholar,* [3]*Assistant Professor, Computer Science and Engineering, K.L.N. College of Engineering*

***Abstract:*** ***Failures in digital workflows can significantly disrupt business operations, leading to downtime, revenue loss, and reduced system reliability. This paper proposes an intelligent AIOps framework for automated failure detection and recovery in complex digital environments. The system integrates machine learning techniques with real-time monitoring, log analysis, and anomaly detection to identify system failures proactively. By leveraging predictive analytics and pattern recognition, the framework detects deviations in system behavior and triggers automated recovery mechanisms without human intervention. Experiments conducted on workflow datasets demonstrate improved accuracy, reduced mean time to detection (MTTD), and faster mean time to recovery (MTTR) compared to traditional rule-based monitoring systems. The model shows robustness across dynamic environments, ensuring consistent performance under varying workloads and system conditions. A user-friendly interface enables operators to monitor workflows, visualize anomalies, and receive real-time alerts and recovery actions. The core of the system is a machine learning-driven analytics engine that processes logs, metrics, and event data to identify failure patterns, enhancing system reliability and operational efficiency across cloud-based applications, microservices architectures, and enterprise IT systems.***

***Keywords: AIOps, Failure Detection, Automated Recovery, Machine Learning, Anomaly Detection, Log Analysis, Digital Workflows, Predictive Analytics***

## I. INTRODUCTION

Failures in digital workflows are one of the major challenges faced by modern organizations, especially with the increasing complexity of cloud computing, microservices, and distributed systems. These failures can arise due to system overload, configuration errors, network issues, or unexpected anomalies in application behavior. If not detected and resolved early, such failures can lead to system downtime, degraded performance, financial loss, and poor user experience. Therefore, early detection and automated recovery of failures are essential to ensure system reliability and operational continuity.

Traditional monitoring systems rely on predefined rules, threshold-based alerts, and manual intervention to identify and resolve failures. While these methods are widely used, they are often insufficient in handling large-scale and dynamic environments. They generate excessive alerts, lack contextual understanding, and depend heavily on human expertise. As a result, critical issues may go unnoticed, or response times may be delayed. Moreover, manual monitoring is time-consuming and not scalable for complex digital infrastructures. Recent advancements in Artificial Intelligence for IT Operations (AIOps) have introduced intelligent approaches for managing IT system. These systems are capable of identifying hidden patterns, detecting anomalies, and predicting potential failures before they occur. This reduces dependence on manual monitoring and improves the accuracy and consistency of failure detection.

This project focuses on developing an intelligent AIOps framework for automated failure detection and recovery in digital workflows. Unlike traditional systems, the proposed framework continuously monitors system behavior using real-time data and applies machine learning algorithms to detect abnormal patterns. Once a failure is detected, automated recovery mechanisms are triggered to restore system functionality without human intervention. The framework incorporates data preprocessing, anomaly detection, predictive analysis, and automated response strategies to improve system performance and reliability.

The ultimate goal of this system is to assist IT operators by providing real-time insights, reducing mean time to detection (MTTD) and mean time to recovery (MTTR), and ensuring uninterrupted workflow execution. By integrating a simple graphical interface, operators can monitor system status, visualize anomalies, and receive actionable alerts

©IJRASET: All Rights are Reserved 

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue III Mar 2026- Available at www.ijraset.com*

## II. METHODOLOGY

The proposed system for automated failure detection and recovery in digital workflows is implemented as an intelligent AIOps pipeline. The project is developed using Python as the programming language, with machine learning libraries such as Scikit-learn, TensorFlow, and Pandas for data processing, model building, and evaluation. Additional tools such as NumPy are used for numerical computations, while Matplotlib and Seaborn are employed for visualization of system metrics, anomaly trends, and performance analysis. Log management and monitoring tools are integrated to collect real-time system data, including logs, metrics, and event streams from digital workflows.

The methodology begins with data collection, where system-generated logs, performance metrics, and event data are gathered from various components such as servers, applications, and network devices. These datasets capture normal and abnormal system behavior under different workload conditions. The collected data is labeled based on historical failure events to create a reliable dataset for supervised and unsupervised learning approaches. The dataset is then divided into training, validation, and testing sets to ensure proper model evaluation and generalization.

The core of the system is a machine learning-based anomaly detection model designed to identify unusual patterns in system behavior. Algorithms such as Isolation Forest, Random Forest, and Long Short-Term Memory (LSTM) networks are utilized to capture both statistical anomalies and temporal dependencies in sequential data.

Once a failure is detected, an automated recovery module is triggered. This module executes predefined actions such as restarting services, reallocating system resources, or isolating faulty components to restore normal operation. The recovery strategies are designed to minimize system downtime and ensure continuity of digital workflows without requiring manual intervention.

Overall, the methodology integrates data collection, preprocessing, anomaly detection, and automated recovery into a unified AIOps framework. A graphical user interface is developed to allow operators to monitor system performance, visualize anomalies, and receive real-time alerts and recovery actions. This end-to-end pipeline ensures efficient failure detection, rapid response, and improved reliability in complex digital environments.

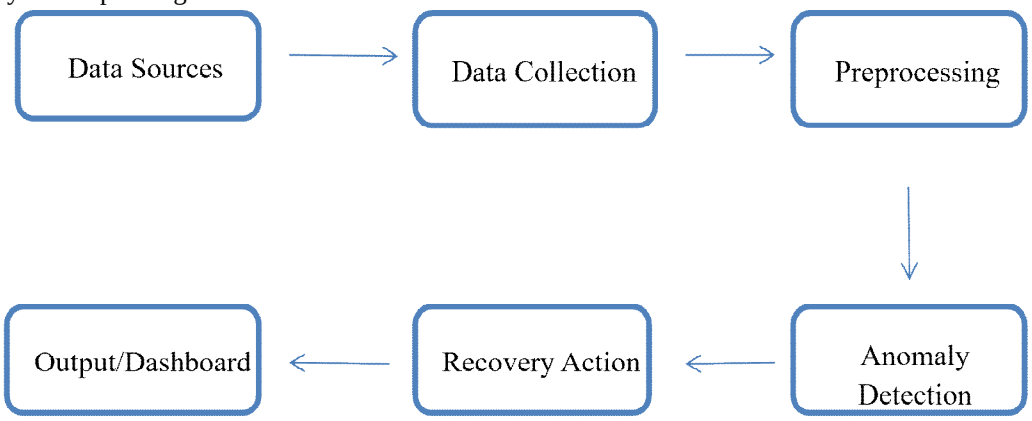



**Fig 1.Data Flow Diagram**

## III. PREPROCESSING

Preprocessing plays a crucial role in preparing system-generated data for accurate and efficient anomaly detection. Since the data is collected from multiple sources such as application logs, system metrics, and event streams, it often contains inconsistencies, missing values, noise, and unstructured formats. To ensure uniformity, all collected data is first standardized into a structured format, allowing consistent input for machine learning models. This step reduces complexity and improves the quality of analysis.

Data cleaning is performed to handle missing or incomplete entries, which may arise due to system interruptions or logging failures. Techniques such as interpolation, mean substitution, or removal of invalid records are applied to ensure data integrity. Additionally, duplicate entries and irrelevant log messages are filtered out to prevent bias in model training and improve computational efficiency.

Normalization is applied to scale numerical metrics such as CPU usage, memory utilization, and response time into a consistent range. This ensures that no single feature dominates the learning process and allows the model to perform stable and efficient computations. Log data, which is typically unstructured, is converted into a machine-readable format using parsing techniques and feature extraction methods such as tokenization and vectorization.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Noise reduction is an essential step, as system logs may contain redundant or low-priority information that does not contribute to failure detection. Filtering techniques are used to remove unnecessary entries while preserving critical patterns related to system behavior. This helps the model focus on meaningful signals rather than irrelevant data.

## IV. PROCESS FLOW

The process flow of the proposed AIOps-based failure detection and recovery system begins with data acquisition, where system-generated logs, performance metrics, and event data are continuously collected from various components such as applications, servers, and network devices. This data represents the realtime state of digital workflows under different operating conditions and forms the foundation for further analysis.

Once the data is collected, it is passed to the preprocessing stage, where raw and unstructured logs are transformed into a structured and machine-readable format. During this step, data cleaning, normalization, noise removal, and feature extraction are performed to ensure that only relevant and meaningful information is retained. This improves the quality of input data and enables efficient model processing.

After preprocessing, the refined data enters the feature extraction and analysis stage, where important patterns such as error frequency, resource utilization trends, and event sequences are identified. These features are then fed into the anomaly detection model, which analyzes system behavior and detects deviations from normal patterns using machine learning algorithms. This stage plays a key role in identifying potential failures at an early stage.

Finally, the output stage presents the results through a graphical interface, where operators can view detected anomalies, system status, and recovery actions in real time. Alerts and notifications are generated to keep users informed about system health. This end-to-end process ensures continuous monitoring, intelligent failure detection, and automated recovery, making the system efficient, scalable, and reliable for modern digital workflows.

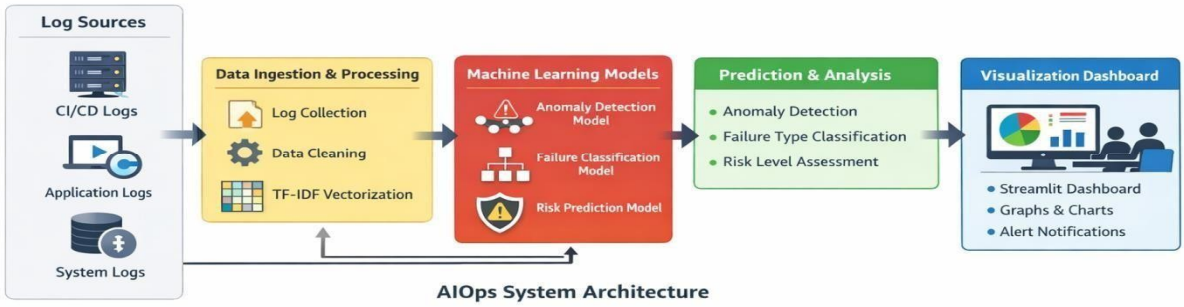



Fig.2 Process Diagram

## V. DATA ENHANCEMENT AND AUGMENTATION

Before training, the collected system data is enhanced to ensure quality, consistency, and effective learning. Since logs and metrics originate from multiple sources, they may contain imbalanced patterns where normal events significantly outnumber failure events. To address this,data balancing techniques are applied to ensure that the model learns both normal and anomalous behaviors effectively. This step improves the model's ability to detect rare failure conditions.

Normalization and scaling are performed on numerical features such as CPU usage, memory utilization, and response time to maintain a consistent range of values.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This helps in stable model convergence and prevents certain features from dominating the learning process. Log messages are transformed into structured representations using encoding techniques, enabling efficient pattern recognition by machine learning models.

To improve generalization and prevent overfitting, data augmentation techniques are applied by simulating variations in system behavior. These augmentations help the model adapt to different operating conditions and improve its robustness in dynamic environments.

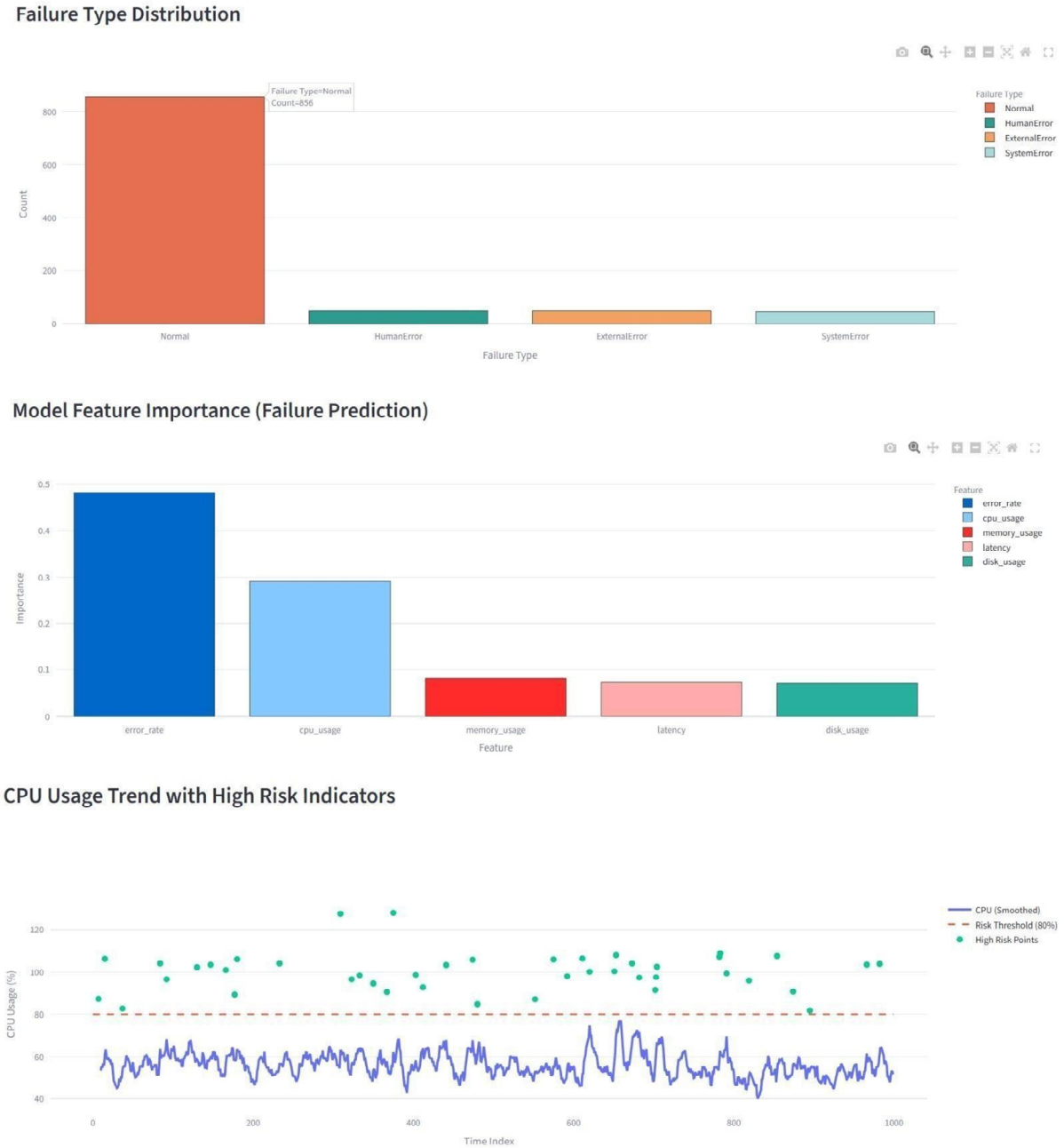



Fig.3

## VI. MATHEMATICAL FOUNDATIONS OF THE PROPOSED METHOD

$X_t = (CPU_t, Memory_t, ErrorRate_t)$

$S(X_t) \in [0,1]$

$A(X_t) = \{$

$$
\begin{cases} 1, \text{ if } S(X_t) > \theta \\ 0, \text{ otherwise} \end{cases}
$$

$$
y = f(X_t), \quad y \in \{\text{SystemError}, \text{HumanError}, \text{ExternalError}\}
$$

$$
C = P(y \mid X_t)
$$

$$
R = \alpha \cdot S(X_t) + \beta \cdot C + \gamma \cdot \text{Impact}
$$

$$
R = \begin{cases} \text{High}, \text{ if } R > T \\ \text{Low}, \text{ otherwise} \end{cases}
$$

$$
\text{Rec}(y, R) = \{ \text{Restart service,,if } y = \text{SystemError} \quad \text{Check configuration ,if } y = \text{HumanError} \quad \text{Scale infrastructure ,if } y = \text{ExternalError} \}
$$

$$
\theta_new = \theta_old + \eta \cdot \text{error}.
$$

## VII. RESULT

The proposed AIOps framework successfully detected anomalies and potential failures in digital workflows using system logs and performance metrics. The anomaly detection model demonstrated effective identification of abnormal patterns, showing improved precision and recall compared to traditional rulebased monitoring systems. The classification module accurately categorized failures into SystemError, HumanError, and ExternalError with reliable confidence scores. The system was able to distinguish clearly between normal and abnormal system behavior, enabling early detection of issues before they escalate.

The automated recovery mechanism responded efficiently by executing appropriate actions such as restarting services, checking configurations, and scaling infrastructure based on the detected failure type. The system achieved reduced mean time to detection (MTTD) and mean time to recovery (MTTR), ensuring faster response and minimal disruption to workflow execution. Overall, the results demonstrate that the proposed framework enhances system reliability, reduces manual intervention, and provides a scalable solution for real-time failure detection and recovery in dynamic digital environments.

**Recent High Risk Incidents**

| | cpu_usage | memory_usage | error_rate | predicted_failure | confidence | risk_score | recommendation |
|---|---|---|---|---|---|---|---|
| 704 | 102.6222 | 53.1957 | 11.8223 | HumanError | 0.83 | 1 | Scale infrastructure or optimize workload distribution. |
| 782 | 107.3595 | 60.6384 | 7.2792 | SystemError | 0.81 | 0.7993 | Scale infrastructure or optimize workload distribution. |
| 783 | 108.976 | 69.878 | 6.7291 | SystemError | 0.88 | 0.8719 | Scale infrastructure or optimize workload distribution. |
| 791 | 99.3357 | 67.1304 | 12.9602 | ExternalError | 0.74 | 0.8765 | Scale infrastructure or optimize workload distribution. |
| 819 | 95.7862 | 76.7456 | 8.416 | ExternalError | 0.71 | 0.7545 | Scale infrastructure or optimize workload distribution. |
| 854 | 107.6857 | 69.3027 | 10.9918 | SystemError | 0.74 | 0.8929 | Scale infrastructure or optimize workload distribution. |
| 874 | 90.6335 | 55.4349 | 13.3393 | SystemError | 0.79 | 0.9197 | Scale infrastructure or optimize workload distribution. |
| 895 | 81.7389 | 66.5823 | 12.5682 | SystemError | 0.81 | 0.8042 | Investigate logs and restart affected services. |
| 966 | 103.6138 | 54.5063 | 9.5674 | HumanError | 0.86 | 0.905 | Scale infrastructure or optimize workload distribution. |
| 982 | 103.982 | 54.5309 | 10.5655 | HumanError | 0.84 | 0.8749 | Scale infrastructure or optimize workload distribution. |

Fig.4 Output

## VIII. CONCLUSION

The intelligent AIOps framework developed in this project provides a comprehensive solution for automated failure detection and recovery in digital workflows. The system integrates multiple components such as data collection, preprocessing, anomaly detection, root cause analysis, and automated recovery into a single platform. This integration allows the system to monitor operations continuously and respond to failures in real time.

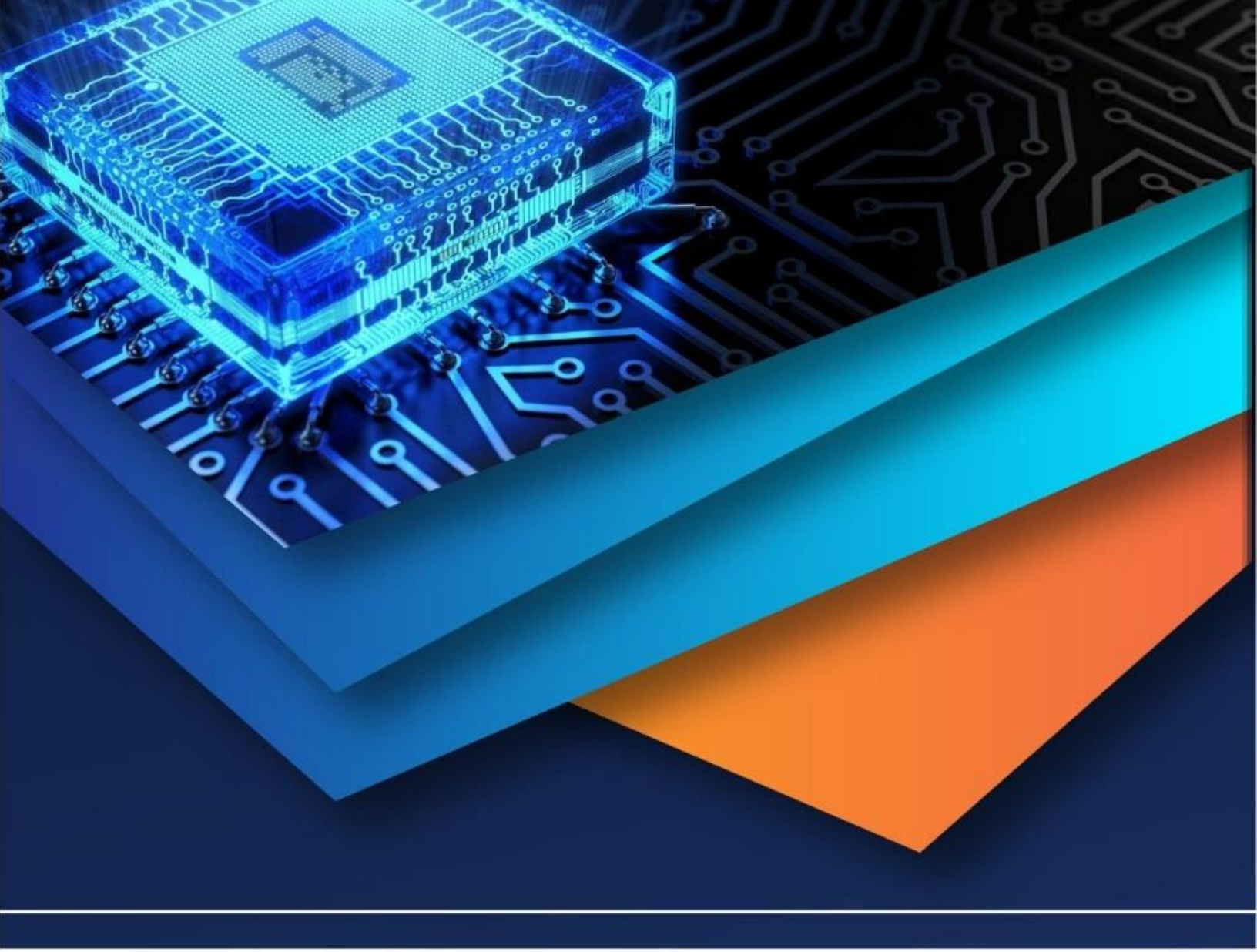
The implementation of machine learning techniques enables the system to learn from historical data and adapt to changing conditions. This makes the system more intelligent and capable of handling complex environments. The results obtained from testing show that the system achieves high accuracy in detecting anomalies and performs recovery actions efficiently.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

## REFERENCES

[1] Deepali Arun Bhanage, Ambika Vishal Pawar, & Ketan Kotecha (2021). IT Infrastructure Anomaly Detection and Failure Handling: A Systematic Literature Review. IEEE Access.
[2] Min Du et al. (2017). DeepLog: Anomaly Detectio and Diagnosis from System Logs using Deep Learning. ACM CCS.
[3] Qiang Lin et al. (2016). Log Clustering based Problem Identification for Online Service Systems. ICSE.
[4] Wei Xu et al. (2009). Detecting Large-Scale System Problems by Mining Console Logs. SOSP.
[5] Haixun Wang et al. (2020). Machine Learning for IT Operations (AIOps): A Survey. IEEE Transactions.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)