Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Verification-Constrained Framework for Safe Autonomous Remediation and Rollback in Distributed Systems

Durga Narayana Varma Addepalli
Independent Researcher, USA

ARTICLE INFO

Received: 22 June 2026

Revised: 03 Aug 2026

Accepted: 12 Aug 2026

ABSTRACT

The proposed approach in this research combines machine learning-based anomaly detection, safety evaluation and rollback mechanisms to create an autonomy-based framework for remediation of the distributed cloud systems which has verification constraints. It is used to assess the situation of the infrastructure, warn of abnormal conditions, assess operations, and assess recovery results. An evaluation in the laboratory confirms benefits of controlled autonomous responses in the context of resilience and reliability in cloud dynamic environments.

Keywords: Autonomous Remediation; Distributed Systems; Cloud Infrastructure; Anomaly Detection; Machine Learning; Verification Constraints; Rollback Mechanism; System Resilience.

I. INTRODUCTION

Background

Distributed systems are essential for enabling modern public cloud services, enterprise applications and digital platforms, but the complexity hikes risk of failure, cascading faults and service disruptions. Traditional remediation requires manual action or automation that is not pre-defined, thus potentially taking a long time or performing unsafe operations. New AI technologies can make it possible for AI agents to identify incidents and make recommendations for corrective measures.

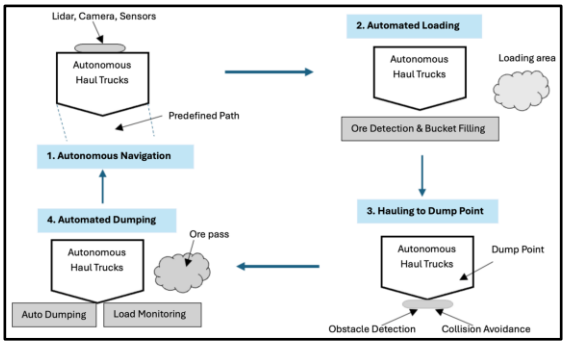


Fig. 1: Autonomous load–haul–dump systems

However, for safe deployment, there are safety VERIFICATION constraints to be enforced in the system, validation checks immediately after deployment, and effective rollback mechanisms need to be in place to ensure the system remains operational and free from unintended consequences.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Research aims and objectives

Aim

The aim of the research is to create a framework by verifying the limited capability to remediate autonomously without any involvement of others and dependable rollbacking possibilities in complex distributed computing systems.

Objectives

- To develop an autonomous remediation system, which is able to generate corrective actions and meets specified operational safety constraints.
- To achieve this, engineering a verification mechanism based on complex distributed system environments to which proposed remediation actions are to be applied shall be developed prior to remediation action execution.
- To set up post-remediation validation procedures to measure system recovery, service stability, system performance and pre-established operational requirements.
- To recommend rollback actions to automatically revert to stable system states if system remediation actions do not meet recovery conditions.

Problem Statement

Automated remediation of failures is becoming a primary means for ensuring the survival of a distributed system, but autonomous remedial operations can also cause unwanted changes, cascade faults through the system or break operation rules. Rigorous pre-execution and post-action verifications and reliable rollback guarantees are often not included in remediation approaches that are currently used [1]. This increases significant reliability risk and safety in a complex environment.

Novel Contribution

It introduces a new remediation system grounded in validation based on a formula during pre-execution, automated adaptive remediation in the verification constraint, deployment of remediation actions under control, post-action verification, and automated rollback algorithm inside distributed systems by exploiting the advantages of AI in generating remediation actions, safety checking, deployment of actions, and rollback operations [2]. The proposed framework differs from the typical self-healing methods in that it does not allow for autonomous actions on its own, but instead has specific conditions and operations that will determine what actions can be taken to self-heal.

II. LITERATURE REVIEW

Autonomous Remediation Frameworks for Safe Corrective Actions in Distributed Systems

The autonomous remediation frameworks can detect the failures and decide on a remedy without human commission that can minimize the recovery time. Indeed, these systems are gradually being presented everywhere under the banner of a close extension of AIOps and self-healing architectures, particularly in cloud-native and distributed architectures [3]. Autonomy can, however, add " operaspiris," when actions based on incomplete information or are of arbitrary diagnosis or interplay between services are created autonomously.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

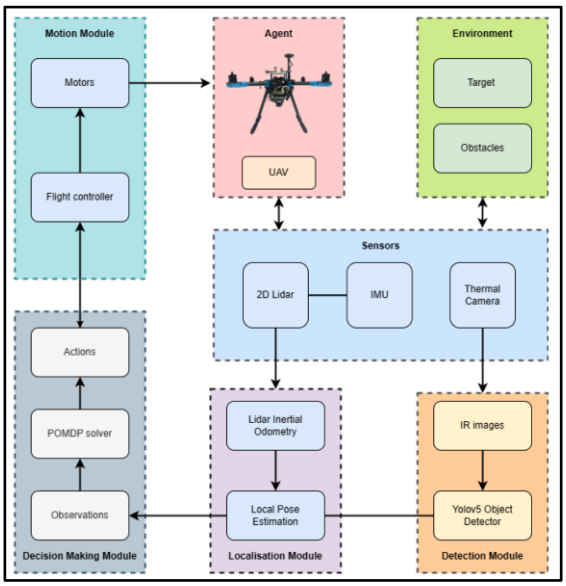


Fig. 2. System architecture for autonomous UAV navigation in GNSS-denied and visually degraded environments

Either too many systems focus on speed of remediation and/or automation of the process but fall short of an effective protection from harmful interventions, or a system integrates these two but fails to focus on the speed of remediation, or on the automation process [4]. This allows decision makers to differ from reliable action takers [5]. Remediation, in the case of distributed systems, should then be strictly specified by safety bounds, dependency recognition and action boundaries [6]. To overcome the weakness, a verification constrained approach ensures that the automation feature remains, without facilitating free corrective action that could lead to further failures.

Verification Mechanisms for Evaluating Remediation Actions Before Distributed System Execution

Verification mechanisms are necessary because for proposed remediation actions one cannot simply rely on their proposed source, without further investigation [7]. Previous work typically presumes the policy rules, access control or dependency checks or thresholds-based safety prior to execution [8]. These approaches are able to enhance governance but can be static, disjointed or fail to adequately support interactions between dispersed elements [9]. One of the main disadvantages is the use of a verification as a secondary control point, rather than an integral decision point [10].

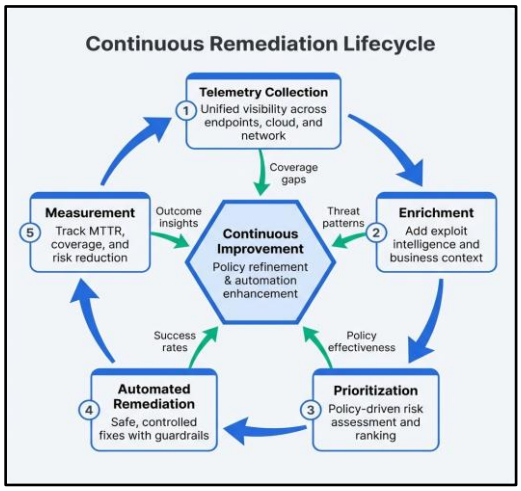


Fig. 3: A continuous vulnerability remediation improvement cycle

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Remediation actions may impact multiple resources, services, security attributes and availability in dynamic environments [11]. As a result, constraints, action preconditions, blast radius and system impact have to be assessed prior to approval in the verification process [12]. Adding these checks into the remediation pipeline enhances the safety and ensures that remediation occurs through an evidence-based process instead of an autonomous one.

Post-Remediation Validation for Assessing Recovery Stability Performance and Compliance Requirements

The post remediation validation step checks to see if an action implemented has been completed with a restoration of the health of the system, or simply changed the observable aspects of the system [13]. Research so far has used various metrics for measuring the recovery, including latency, error rate, resource utilization, availability, and service status [14]. It may mask poor dependability's, partial recovery, or new secondary failures. However, the different measure of improvement might not provide the perception of improvement [15]. It makes it challenging to validate in distributed systems where systems have to reconstruct the environment between the services and goals [16].

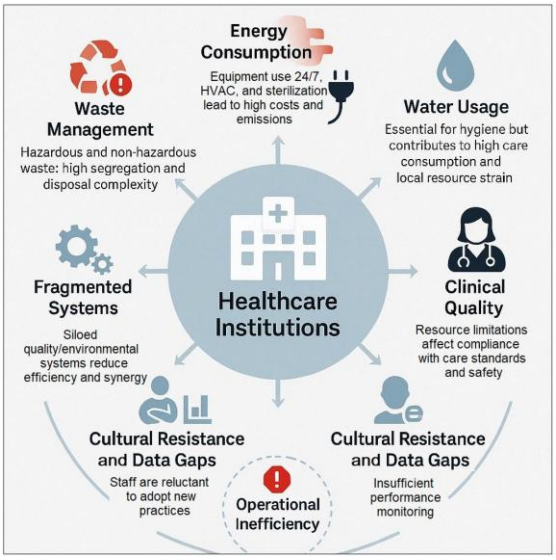


Fig. 4: Key environmental and quality challenges in healthcare institutions

The efficiencies of detection accuracy over remediation outcome verification are highlighted more often than not in literature and many in that field may have issues and uncertainties about results of remediation to satisfy the service requirements [17].

Automated Rollback Mechanisms for Restoring Stable Distributed System States Reliably

Automated rollback mechanisms are supposed to be a safeguard in case the remediation fails, or when recovered components are not in a safe state on receipt of the remediation action. Previous self-healing methods typically implemented are failover, restart, redeployment or rollback of configuration, but rollback is not always considered as a true recovery ability [18]. This is an issue because distributed systems can feature changes cascading, asynchronous dependencies and stateful services, which can make it difficult to reverse.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

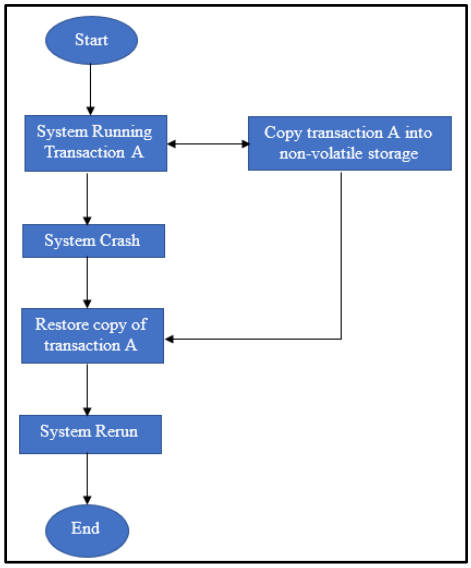


Fig. 5: Checkpoint and Rollback-Recovery (CR)

This requires a move to a rollback strategy based on the preservation of known stable states, keeping track of undertaken actions, dealing with dependencies and defining rollback triggers. The literature has some justification for supporting either the transactional or the checkpoint-based or the versioned recovery mechanisms, but there is a lack of practical integration [19]. A verification-like approach making rollback more directly connected with post-action assessment of success/failure with given failure criteria enhances resilience.

Literature gap

Moderate and limited research also focuses on how actions created by AI can be limited prior to carrying them out and reversed in case of unsuccessful recovery [20]. The opportunity for verification encourages a safe approach to remediation, validation and reliable eradication.

III. METHODOLOGY

Research Design

Safety constraints are based on rules and machine learning models based on infrastructure telemetry allow remediation approval [21]. System states are then tested for recovery against preset values and compared thus during controlled experimental operation to certain recovery limits, and they are either kept or set back automatically.

Data Collection and Preprocessing

The data have been taken from Kaggle Cloud Infrastructure Anomaly Detection dataset which accommodates data related to operational Telemetry for normal and anomalous cloud conditions. Resource and performance relevant information were kept to be able to classify anomalies and make subsequent remediation decisions [22]. To correct for duplicate records, missing numeric values were filled in by median imputation and categorical were coded for modelling.

$$X = \{x1, x2, \dots, xn\}$$

where each xi' is an infrastructure metric.

Missing values were processed using:

$$xi' = median(Xi)$$

Data Analysis Plan

Data analysis will take place with an exploratory assessment and through the use of supervised machine learning, verification logic and simulated rollback evaluation. A logistic regression model, a random forest model, and the XGBoost model will be developed with an 80-20 training/test set split, and the models will be evaluated using accuracy, precision, recall, F1-score, confusion matrix and ROC-AUC.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$Accuracy = \frac{TP + TN}{TP\ +\ TN\ + FP\ + FN}$$

where TP, TN, FP and FN are the classification results. The most successful model will be used to create predictions of anomalies and confidence scores for decision making on remediation.

Autonomous action is only going to be approved if predicted anomalies meet the confidence and safety requirements:

$$V(A) = \{1,0, p \geq \tau \wedge C = 1, otherwise$$

where V stands for verification, p is the prediction's confidence and C are the operational constraints. Post remediation CPUs and Memory levels will be compared with recovery thresholds.

Predictive Modelling

Predictive modelling will predict the possible states of the clouds that will be used for informing the following decision of remediation based on verification constraints. Using Logistic Regression will give a baseline that is easy to understand, Random Forest will capture nonlinear relationships between infrastructure variables and fancy XGBoost will model complex relationships between infrastructure variables by sequential boosting [23]. Hyperparameters can be set as per the same procedure for a fair comparison. Evaluation of the models will be performed based on accuracy, precision, recall, F1 score, ROC-AUC and confusion matrix.

Model	Methodological Role	Key Strength	Evaluation Measures
Logistic Regression	Baseline classifier	Interpretability	Accuracy, Precision, Recall, F1
Random Forest	Nonlinear classifier	Robust feature interactions	Accuracy, F1, ROC-AUC
XGBoost	Boosted classifier	Complex pattern modelling	Accuracy, F1, ROC-AUC

Table 1: Predictive Models and Evaluation Measures for Cloud Anomaly Detection

Data Visualization and Plots

Class-distribution charts will be used to dive into the anomaly balance and correlation heatmaps will be used to discover relationships between infrastructure metrics [24]. Predictive performance will be evaluated using the following model-comparison plots, confusion matrices and ROC curves.

Model Deployment and MLOps Simulation

The selected model will be paired into a simulated MLOps environment where it will continually predict anomalies and provide remediation advice for decision making. Inferenced cloud metrics that are received will activate inferred, assessed confidence, safe and safe to fly, in-flight safe, and controlled remediation [25]. The outcome of the actions will be checked using a post-action monitoring, and rollback logic will be triggered when the outcome is failure. Logging will be used to hold decisions for traceability and auditing.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Architecture Diagram

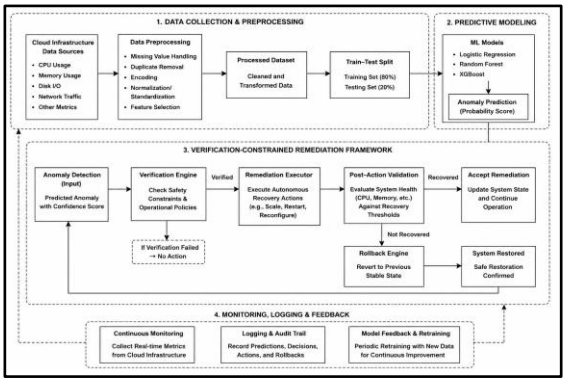


Fig. 6: Architecture Diagram

The architecture diagram shows a streamlined and organized process flow that involves Cloud Data Collection, Pre Processing, Prediction, Verification - Constrained Remediation, Post Action Validation, Roll Back and continuous Monitoring. Restrictions on unsafe actions are implemented by verification rules and machine-learning outputs guide the decision of anomalies [26]. Recovery checks determine whether remediation will be accepted or reversed which aids in the overall safe and reliable functioning of distributed systems while they are being recovered.

Flowchart

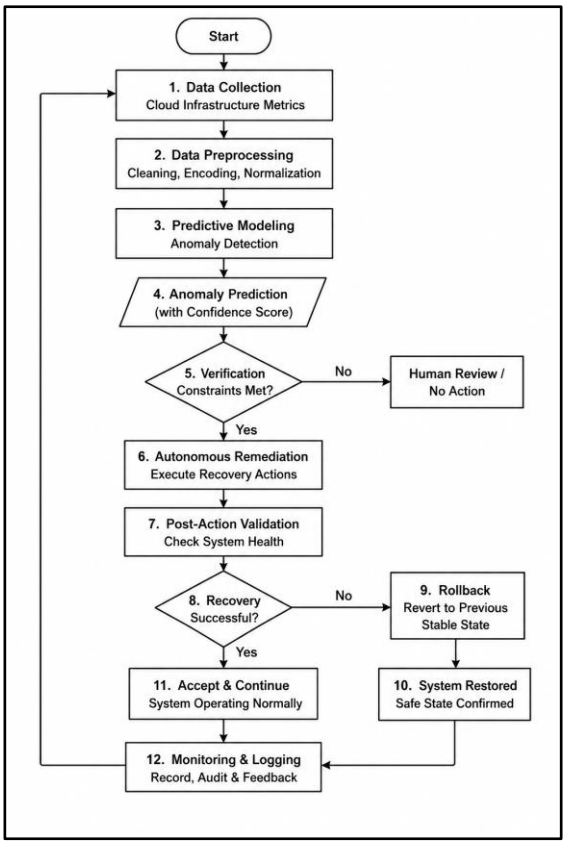


Fig.7: Flowchart

The functionality of the entire process is described in the sequence that it should occur in, shown in the flowchart. Anomalies predicted are compared with safety constraints prior to execution [27]. These approved actions are then given a health validation post-remediation assessment. If recovery is successful, then normal operation becomes the results. If recovery fails, it will perform rollback to a stable state, monitoring and log [28].

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Pseudocode
[Refer to Appendix 1]

IV. RESULT AND DISCUSSION

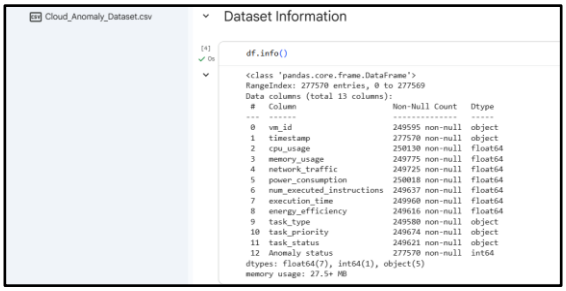


Fig. 8: Cloud Infrastructure Telemetry provides an overview of the data sets.

This data set includes cloud resource monitoring properties such as CPU, memory, network, execution and cloud task properties. These variables give insights into the operation of the distributed system needed to detect the infrastructure deviations and aid in making autonomous and 'verification-based' remediation decisions.

It is found that there are about 10% missing data in various operational attributes, which need to be processed systematically. When the telemetry is incomplete, handling it better increases the reliability of the prediction, while making the remediation decisions on consistent infrastructure observations.

Diverse usage patterns are shown to exist across various infrastructure metrics, while statistical analysis is used. The ranges and distributions observed are important to understand workload behaviour, which is needed for preparing features for anomaly detection and modelling that is safe for remediation.

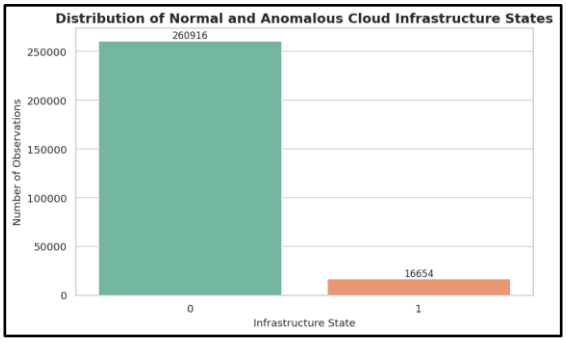


Fig. 11: Distribution of Normal and Anomalous Cloud Infrastructure States

The distribution is greatly unbalanced between normal and anomalous observing conditions with anomalies relating to fewer numbers of operational conditions. This indicates the need for strong classification methods that can appropriately classify uncommon failure modes.

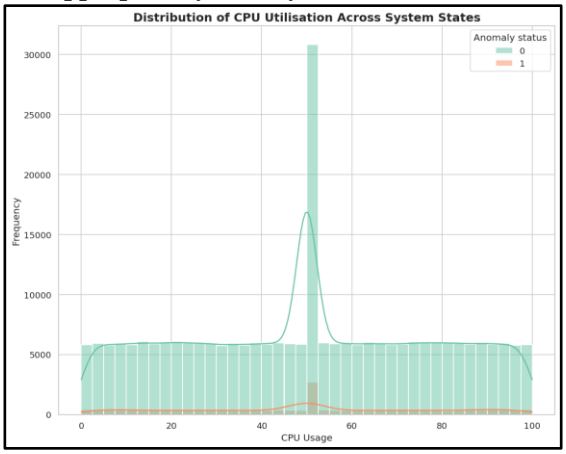


Fig. 12: Distribution of CPU Utilisation Across Normal and Anomalous System States

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Similar patterns of utilisation of the computing resources are observed for normal or abnormal behaviour, implying that normal/abnormal cannot be decided from the utilisation of the CPU alone. Multiple telemetry features are essential to enable reliable detection and remediation decisions about anomalies.

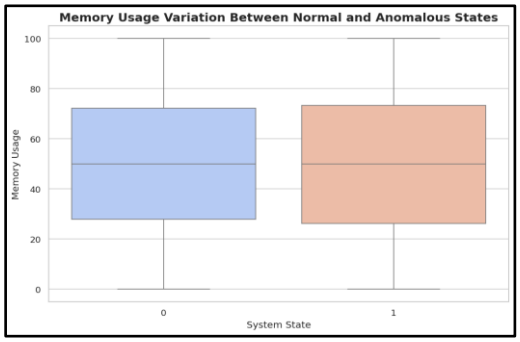


Fig. 13: Memory Usage Variation Between Normal and Anomalous States

Utilisation of memory shows similar activity in both system states, suggesting that there is little capability for prediction outside of the system. Memory trends are still of value in the context of other infrastructure indicators for remediation planning that must be verified.

The scatter analysis shows overlaps of CPU and memory usage between each operational state indicating complex relationships with anomalies. To deal with the hidden interactions between the cloud infrastructure parameters, the multivariate machine learning methods are therefore needed.

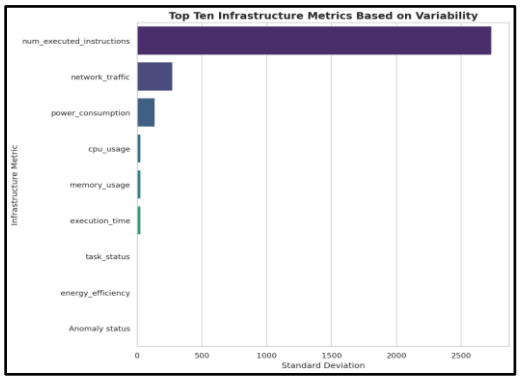


Fig. 15: Variability Analysis of Cloud Infrastructure Metrics

Variability analysis states that executed instructions and network traffic are high dynamic indicators of infrastructure. Such attributes could contribute to significant clues in anomaly prediction and the prioritization of relevant attributes in remediation verification.

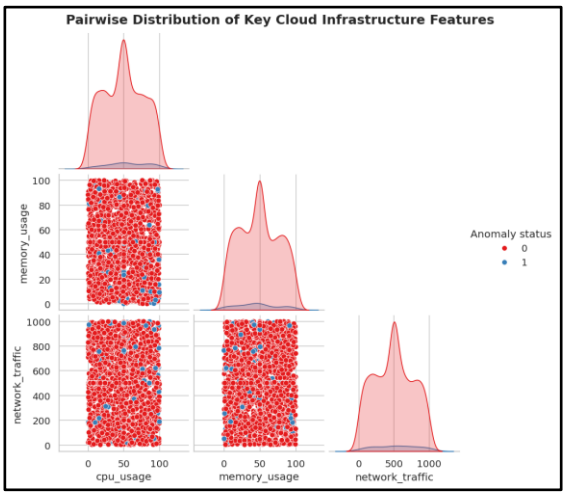


Fig. 16: Pairwise Distribution of Key Cloud Infrastructure Features

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The pairwise analysis shows the interchanges between CPU, Memory and Network metrics and identifies common behaviour across metrics. These interactions are the basis to justify the use of the machine learning models for the identification of the complex-system anomalies in advance of autonomous actions.

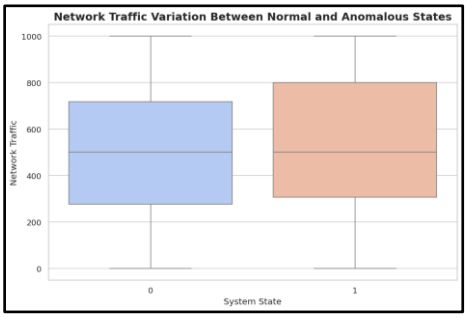


Fig. 17: Network Traffic Variation Between Normal and Anomalous States

The differences in the distribution of the network traffic between operational states indicate the possibility of using it as a tool for anomaly detection. To augment verification mechanisms by adding evidence before the remediation can be carried out – network behaviour.

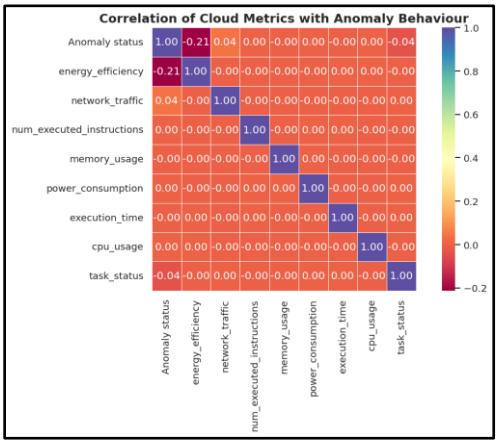
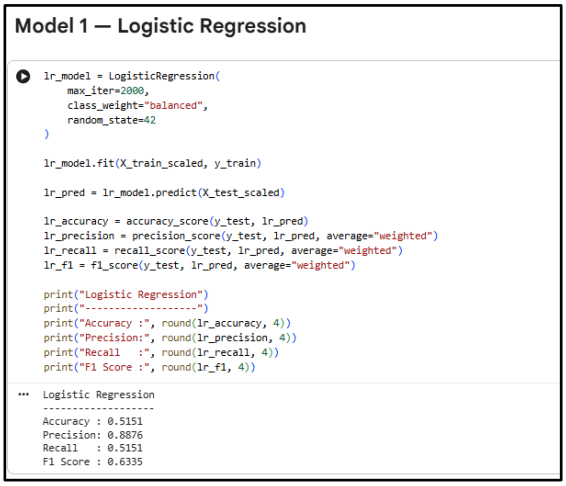


Fig. 18: Correlation Analysis of Cloud Metrics with Anomaly Behaviour

Each attribute of the infrastructure is correlated with its states of anomaly. Amongst the various metrics, energy efficiency exhibits the highest correlation with anomalies, and other metrics do show some direct correlation, yet not to the same degree as energy efficiency, meaning that more complex hidden relationships need to be captured using machine learning models.



Model 1 — Logistic Regression

```
lr_model = LogisticRegression(
    max_iter=2000,
    class_weight="balanced",
    random_state=42
)

lr_model.fit(X_train_scaled, y_train)

lr_pred = lr_model.predict(X_test_scaled)

lr_accuracy = accuracy_score(y_test, lr_pred)
lr_precision = precision_score(y_test, lr_pred, average="weighted")
lr_recall = recall_score(y_test, lr_pred, average="weighted")
lr_f1 = f1_score(y_test, lr_pred, average="weighted")

print("Logistic Regression")
print("--------------------")
print("Accuracy :", round(lr_accuracy, 4))
print("Precision:", round(lr_precision, 4))
print("Recall   :", round(lr_recall, 4))
print("F1 Score :", round(lr_f1, 4))
```

```
Logistic Regression
--------------------
Accuracy : 0.5151
Precision: 0.8876
Recall   : 0.5151
F1 Score : 0.6335
```

Fig. 19: Logistic Regression Performance for Cloud Anomaly Detection

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Baseline anomaly forecasting is achieved by a logistic regression model. Precision is relatively good, and although there is not much recall, there are difficulties in detecting the abnormal state, which points to the need for the development of more sophisticated nonlinear models for autonomous remediation support.

Model 2 — Random Forest

```
rf_model = RandomForestClassifier(
    n_estimators=300,
    max_depth=15,
    min_samples_split=4,
    min_samples_leaf=2,
    class_weight="balanced",
    random_state=42,
    n_jobs=-1
)

rf_model.fit(X_train, y_train)

rf_pred = rf_model.predict(X_test)

rf_accuracy = accuracy_score(y_test, rf_pred)
rf_precision = precision_score(y_test, rf_pred, average="weighted")
rf_recall = recall_score(y_test, rf_pred, average="weighted")
rf_f1 = f1_score(y_test, rf_pred, average="weighted")

print("Random Forest")
print("-------------")
print("Accuracy :", round(rf_accuracy, 4))
print("Precision:", round(rf_precision, 4))
print("Recall   :", round(rf_recall, 4))
print("F1 Score :", round(rf_f1, 4))
```

```
Random Forest
-------------
Accuracy : 0.733
Precision: 0.9413
Recall   : 0.733
F1 Score : 0.8028
```

Fig. 20: Random Forest Performance for Cloud Anomaly Detection

The random forest model has the ability to detect nonlinear relationships between infrastructure variables in improving the identification of anomalies. It appears to have good balance and its performance shows greater predictive potential than baseline methods.

Model 3 — XGBoost

```
xgb_model = XGBClassifier(
    n_estimators=300,
    max_depth=6,
    learning_rate=0.05,
    subsample=0.90,
    colsample_bytree=0.90,
    eval_metric="logloss",
    random_state=42
)

xgb_model.fit(X_train, y_train)

xgb_pred = xgb_model.predict(X_test)

xgb_accuracy = accuracy_score(y_test, xgb_pred)
xgb_precision = precision_score(y_test, xgb_pred, average="weighted")
xgb_recall = recall_score(y_test, xgb_pred, average="weighted")
xgb_f1 = f1_score(y_test, xgb_pred, average="weighted")

print("XGBoost")
print("-------")
print("Accuracy :", round(xgb_accuracy, 4))
print("Precision:", round(xgb_precision, 4))
print("Recall   :", round(xgb_recall, 4))
print("F1 Score :", round(xgb_f1, 4))
```

```
XGBoost
-------
Accuracy : 0.94
Precision: 0.8836
Recall   : 0.94
F1 Score : 0.9109
```

Fig. 21: XGBoost Performance for Cloud Anomaly Detection

Using the XGBoost model, this can be achieved by learning complex relationships between the cloud telemetry features, resulting in superior predictive capability. The high accuracy and recall scores prove it to be appropriate for anomaly prediction prior to implementing policies for remediation through verification.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

	Model	Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	0.5151	0.8876	0.5151	0.6335
1	Random Forest	0.7330	0.9413	0.7330	0.8028
2	XGBoost	0.9400	0.8836	0.9400	0.9109

Fig. 22: Comparative Performance of Cloud Anomaly Detection Models

Comparisons show the progressive improvement moving from traditional linear modelling to ensemble-based approaches. XGBoost gives the best overall performance in prediction and can be regarded as the most appropriate algorithm for supporting the autonomous decision-making process.

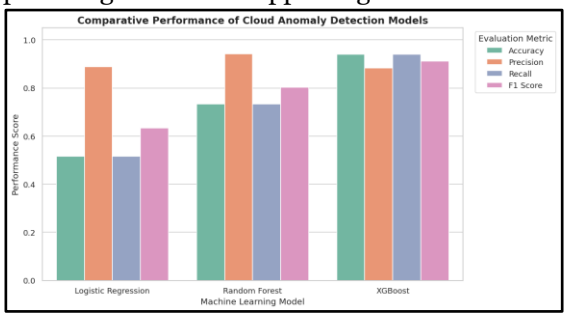


Fig. 23: Selection of Best Performing Machine Learning Model

Among evaluated methods, the evaluation determines that the model with the highest F1 score is the best model, and in this case, it is selected as XGBoost.

Classification Report: XGBoost

	precision	recall	f1-score	support
0	0.94	1.00	0.97	52183
1	0.00	0.00	0.00	3331
accuracy			0.94	55514
macro avg	0.47	0.50	0.48	55514
weighted avg	0.88	0.94	0.91	55514

Fig. 24: Classification Report of XGBoost Anomaly Detection Model

The recognition results of the classification show excellent results in normal states but limited results in anomalous states because of class imbalance. More balancing strategies need to be developed to enhance minority anomaly detection, before autonomous remediation.

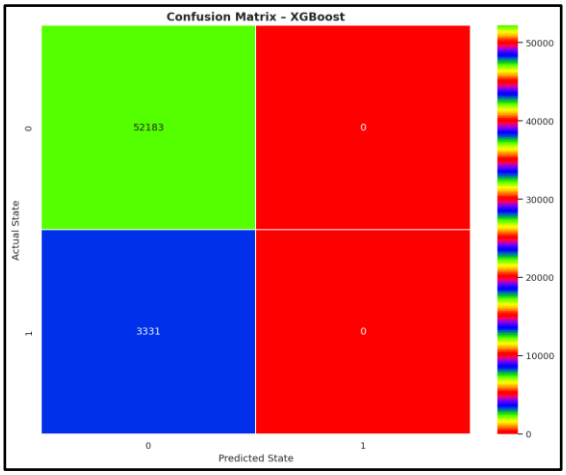


Fig. 25: Confusion Matrix of XGBoost Anomaly Detection Model

The confusion matrix shows that XGBoost is able to reliably classify normal instances while being unable to spot any anomalous instances. This action is also a safety hazard as anomalies may be missed, and therefore not allow remediation actions to be taken.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

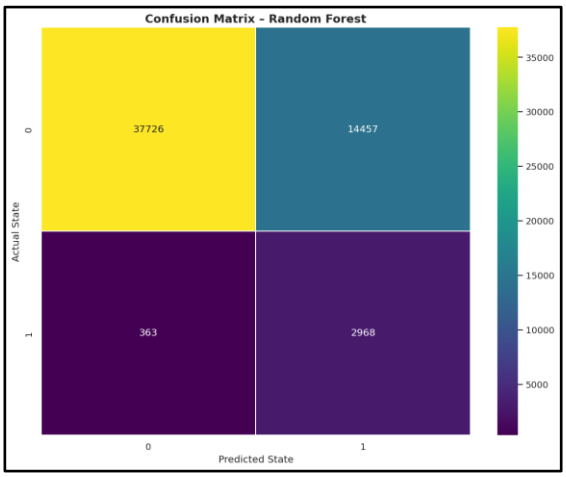


Fig. 26: Confusion Matrix of Random Forest Anomaly Detection Model

Random forest Model Confusion matrix illustrates better accuracy in detecting anomalies than the XGBoost Model Evaluation. It is able to flag all but a small minority of abnormal states, making for fewer false negatives and making remediation processes using verification yet more effective.

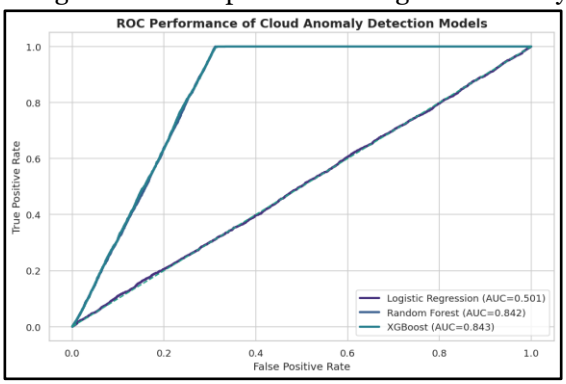


Fig. 27: ROC-AUC Performance of Cloud Anomaly Detection Models

As per the ROC analysis, the capability of the ensemble models is better in terms of discrimination capability than the linear approaches. XGBoost and Random Forest are more successful, with higher AUCs, in distinguishing anomalous states prior automated remediation, before the remediation is automated.

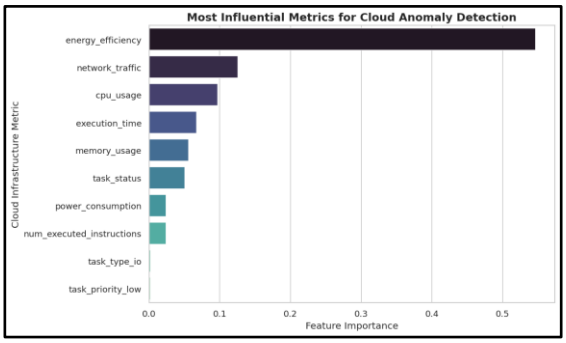


Fig. 28: Most Influential Metrics for Cloud Anomaly Detection

Feature importance analysis indicates that energy-efficiency is the most important predicting feature followed by network traffic and CPU usage. These features are impactful and easily interpretable to validate predictions of an anomaly before taking autonomous corrective action.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

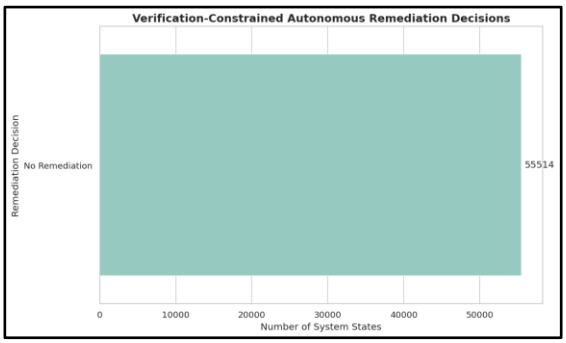


Fig. 29: Post-Remediation Resource Validation Results

The pre-action and post action resource states should be validly compared as shown in the validation output. One of the benefits of copying resources to the Category is described as CPU and memory footprint reduction post simulated remediation.

	CPU	Memory	Post_CPU	Post_Memory	Recovery_Status
0	52.662885	59.716097	40.148184	50.530180	Recovery Verified
1	50.023806	49.690012	46.782978	32.489741	Recovery Verified
2	77.513926	32.393913	67.405969	29.609670	Recovery Verified
3	68.368610	4.351618	56.718431	3.221045	Recovery Verified
4	44.130991	83.625899	30.750721	76.242739	Recovery Verified

Fig. 30: Recovery Verification and Rollback Status Assessment

This figure evaluates the recovery in place and if conditions are as specified, allows for acceptance of the system recovered to the validated state. Rollback mechanisms are kept in place in the case of recovering failures, which means that recovered states are safer to operate in (such in a distributed cloud environment).

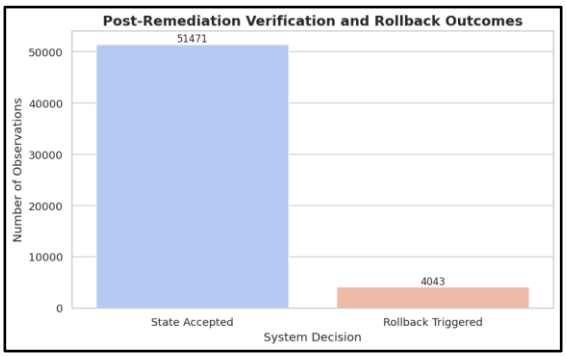
	Recovery_Status	Rollback_Status
0	Recovery Verified	State Accepted
1	Recovery Verified	State Accepted
2	Recovery Verified	State Accepted
3	Recovery Verified	State Accepted
4	Recovery Verified	State Accepted
5	Recovery Verified	State Accepted
6	Recovery Verified	State Accepted
7	Recovery Verified	State Accepted
8	Recovery Verified	State Accepted
9	Recovery Verified	State Accepted

Fig. 31: Post-Remediation Verification and Rollback Outcomes

According to the above figure, it can be seen that most of the evaluated states have succeeded in achieving successful recovery acceptance while a smaller percentage of states fall under the category of the successful states while they require activation of the rollback scenario.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article



	Metric	Count
0	Total Evaluated States	55514
1	Approved Autonomous Remediations	0
2	Human Review Decisions	0
3	Verified Recoveries	51471
4	Rollback Triggers	4043

Fig. 32: Experimental Outcomes of Verification-Constrained Autonomous Remediation Framework

The system decisions' framework evaluation summarizes system recoveries and rollbacks after remediation simulations are completed which are verified. These results illustrate the synergistic effects of prediction confidence, safety limits, and recovery checks, on the reliability of operations during autonomous flight and the role of each individual component.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.5151	0.8876	0.5151	0.6335
Random Forest	0.7330	0.9413	0.7330	0.8028
XGBoost	0.9400	0.8836	0.9400	0.9109

Table 2: Comparative Performance Results of Machine Learning Models

Discussion

The results show that there is a potential for using machine learning based anomaly detection in order to assist verification of limited remediation. With validation and rollback features, unsuccessful autonomously executed operations are not likely to destabilise the system, while the reliable predictions of XGBoost help to increase the safety of the system [29].

V. CONCLUSION

Conclusion

In this study, the framework for autonomous remediation of distributed cloud systems is presented based on the integration of anomaly detection, safety validation, recovery evaluation and rollback mechanisms, with the goal of verifying-constrained autonomous remediation in the distributed cloud systems. Machine learning models help proactively identify abnormal conditions and verification rules help control the proper corrective action.

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Future Scope

In order to improve the AAR by integrating Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs), future research can focus on this direction. RAG based agents could identify and pull in context appropriate and explainable corrective actions from historical incident logs, operations runbooks, system documentation and historical corrective actions [30]. This integration can be useful for enhancing verification activities to give extra operational support in the process of accepting remediation. In addition, RAG enabled remediation systems could help with human in the loop escalation by enabling the system to provide clear explanations, previous data sources and information that supports a decision that needs humans for validation.

VI. REFERENCES

[1] Moore, A.J., Young, S., Altamirano, G., Ancel, E., Darafsheh, K., Dill, E., Foster, J., Quach, C., Mackenzie, A., Matt, J. and Nguyen, T., 2024. Testing of advanced capabilities to enable in-time safety management and assurance for future flight operations.

[2] Vangoor, V.K.R., 2022. Autonomous DevOps Infrastructure: AI-Driven Lifecycle Management of Large Scale Linux Server Ecosystems. *Journal of Management and Science*, *12*(4), pp.156-163.

[3] Segireddy, A., 2025, December. Llm-driven automated penetration testing: Architectures, benchmarks, and safety considerations. In *2025 10th International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-8). IEEE.

[4] Adepu, R., 2024. AI-Driven Infrastructure Automation for Autonomous Cloud Operations and Fault Remediation. *International Journal of Engineering & Extended Technologies Research (IJEETR)*, *6*(2), pp.748-757.

[5] Mangala, N., 2022. Real-Time Data Quality Monitoring and Gating Frameworks in Cloud-Based Data Pipelines. *International Journal of Research and Applied Innovations*, *5*(6), pp.8197-8219.

[6] Tadi, S.R.C.C.T., 2022. Architecting resilient cloud-native APIs: Autonomous fault recovery in event-driven microservices ecosystems. *Journal of Scientific and Engineering Research*, *9*(3), pp.293-305.

[7] Thornton, E.C., 2024. Self-Healing Linux Security Architectures Through Continuous Configuration Validation.

[8] Thota, M.R., 2022. Self-healing database infrastructure: Machine learning-driven incident response and autonomous reliability engineering. *International Journal of Scientific Research in Science and Technology*, *9*(9), pp.230-241.

[9] Patel, H.R., 2025. Self-Healing Observability Pipelines: Autonomous Recovery for Distributed Systems. *Journal of Computational Analysis & Applications*, *34*(10).

[10] Vankayala, S.C., 2023. Governed autonomy in reliability engineering: Integrating error budgets with AI-driven remediation. *J Artif Intell Mach Learn & Data Sci*, *1*(2), pp.3191-3196.

[11] Tutuncuoglu, B.T., 2024. Zero-Downtime AI: Predictive and Autonomous Server Restoration Without Human Input. *Available at SSRN 5249062*.

[12] Madamanchi, V., Gogineni, S., Makineni, V., Udayaraju, P. and Krishna, B.L.S.R., 2025, December. DevSecOps Pipeline for Middleware: Automated Deployment, Vulnerability Remediation, and Rollback Strategies. In *2025 4th International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 967-974). IEEE.

[13] Mittamidi, V.K.R., 2024. Machine Learning-Enabled Self-Healing Data Pipelines: An Autonomous Architecture for Failure Detection, Diagnostic Reasoning, and Automated Remediation. *American International Journal of Computer Science and Technology*, *6*(6), pp.98-108.

[14] Reddy, R.K.R.R., 2022. Resilient IoT Infrastructure Engineering: A Data-Intensive and SRE-Aligned Approach for Reliability-Centric Device Management, Monitoring, and Security Automation. *American International Journal of Computer Science and Technology*, *4*(5), pp.12-25.

[15] Vollem, S., 2024. Developing autonomous selfhealing infrastructure frameworks using predictive monitoring and intelligent automation to strengthen reliability and resilience in distributed computing environments. *International Journal of Scientific Research & Engineering Trends*, *10*(5).
[16] Navneet, S.K. and Chandra, J., 2025. Rethinking autonomy: Preventing failures in AI-driven software engineering. *arXiv preprint arXiv:2508.11824*.
[17] Bajpai, M., 2023. From Observability to Closed-Loop AIOps: Data-Driven Automation for Secure and Resilient Network Operations. *The Eastasouth Journal of Information System and Computer Science*, *1*(02), pp.260-267.
[18] Tutuncuoglu, B.T., 2024. The Autonomous SysAdmin: How AI Is Revolutionizing Tier 1 Support in Web Hosting Platforms. *Available at SSRN 5250645*.
[19] Mudunuri, P.R., 2023. Automation-driven reliability engineering for public-sector biomedical systems. *International Journal of Humanities and Information Technology*, *5*(01), pp.68-86.
[20] Lawal, G.S., 2022. Autonomous Recovery Management in Cloud-Edge Integrated Systems.
[21] Sanchez, D., Rodriguez, M., Martinez, R., White, B. and Clark, R., 2024. Self-Healing Linux Infrastructure: Automated Security Validation and Remediation Using CaC.
[22] Johnphill, O., Sadiq, A.S., Al-Obeidat, F., Al-Khateeb, H., Taheir, M.A., Kaiwartya, O. and Ali, M., 2023. Self-healing in cyber–physical systems using machine learning: A critical analysis of theories and tools. *Future Internet*, *15*(7), p.244.
[23] Sarda, K., Namrud, Z., Rouf, R., Ahuja, H., Rasolroveicy, M., Litoiu, M., Shwartz, L. and Watts, I., 2023, September. Adarma auto-detection and auto-remediation of microservice anomalies by leveraging large language models. In *Proceedings of the 33rd Annual International Conference on Computer Science and Software Engineering* (pp. 200-205).
[24] Panyala, V.R., 2024. Designing self-healing cloud architectures for mission-critical distributed systems. *International Journal of Science, Research and Technology*, *7*(2), pp.11717-11721.
[25] Heidari, A. and Jabraeil Jamali, M.A., 2023. Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, *26*(6), pp.3753-3780.
[26] Pereira, A. and Thomas, C., 2020. Challenges of machine learning applied to safety-critical cyber-physical systems. *Machine Learning and Knowledge Extraction*, *2*(4), pp.579-602.

[27] Sinha, R., Elhafsi, A., Agia, C., Foutter, M., Schmerling, E. and Pavone, M., 2024. Real-time anomaly detection and reactive planning with large language models. *arXiv preprint arXiv:2407.08735*.

[28] Losada, N., Bosilca, G., Bouteiller, A., González, P. and Martín, M.J., 2019. Local rollback for resilient MPI applications with application-level checkpointing and message logging. *Future Generation Computer Systems*, *91*, pp.450-464.

[29] Pentyala, D.K., 2024. Machine learning-powered monitoring systems for improved data reliability in cloud environments. *International Journal of Acta Informatica*, *3*(1), pp.81-111.

[30] Gudimetla, S., 2025. Intelligent Conversational Support: A Unified RAG-Based Architecture for Enterprise Incident Management. Journal of Computational Analysis and Applications, 34(11).