E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

AI-Driven Machine Learning Framework for Proactive Anomaly Detection in Mission-Critical Software Systems

DOI: https://doi.org/10.63345/ejset.v2.i2.301

Chandana Pandey,

Research Scholar,

Dr Gauri Shanker Kushwaha,

Research Supervisor

Maharaja Agrasen Himalayan Garhwal University, Uttarakhand

http://www.ejset.org/ || **Vol. 2 No. 2 (2026): June Issue**

Date of Submission: 03-05-2026	**Date of Acceptance:** 17-05-2026	**Date of Publication:** 20-06-2026

Abstract— Mission-critical software systems operating in domains such as healthcare, aerospace, industrial automation, telecommunications, and cloud computing require highly reliable anomaly detection mechanisms to prevent service disruptions and catastrophic failures. Although recent advances in artificial intelligence (AI) and machine learning (ML) have significantly improved anomaly detection accuracy, existing approaches remain constrained by several limitations, including dependence on single-source monitoring data, limited adaptability to concept drift, inadequate support for real-time prediction, poor model interpretability, and insufficient integration with operational decision-making. Moreover, most current solutions focus on identifying anomalies after they have occurred rather than enabling proactive system resilience through early warning and intelligent risk assessment. This research proposes an AI-Driven Machine Learning Framework for Proactive Anomaly Detection in Mission-Critical Software Systems that addresses these shortcomings through a unified, multi-layered architecture. The proposed framework integrates heterogeneous operational data, including system logs, performance metrics, execution traces, and runtime events, to provide a comprehensive representation of system behaviour. It incorporates adaptive machine learning models capable of continuously learning from evolving software environments while mitigating concept drift through incremental model updates. To improve operational trust and decision support, the framework further combines explainable AI techniques with a risk-aware anomaly prioritization mechanism that assists system administrators in distinguishing critical anomalies from benign deviations. Unlike conventional anomaly detection systems that primarily perform retrospective fault identification, the proposed framework emphasizes proactive anomaly forecasting, intelligent alert prioritization, and early intervention to enhance software resilience. The study presents the conceptual architecture, methodological design, and evaluation strategy of the proposed framework using publicly available benchmark datasets and standard anomaly detection metrics, including Precision, Recall, F1-score, Area Under the ROC Curve (AUC), detection latency, false alarm rate, and computational efficiency. The anticipated contribution of this research is the development of an integrated AI-driven framework that bridges the gap between accurate anomaly detection and practical deployment in mission-critical software environments by combining adaptive learning, multimodal data fusion, explainable decision-making, and proactive operational intelligence. The proposed framework aims to improve the reliability, scalability, and resilience of next-generation software systems while providing actionable insights for real-time system management.

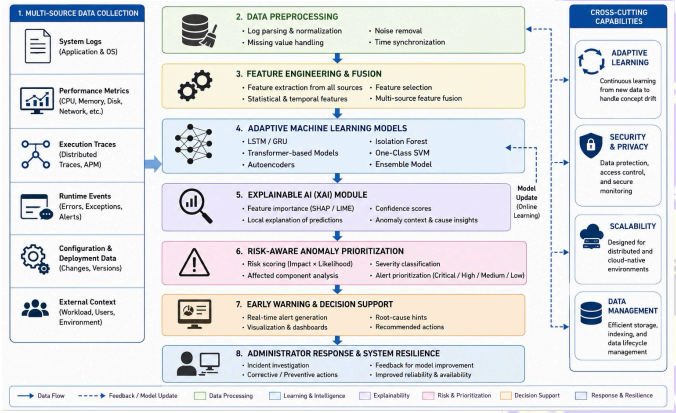
Keywords— Artificial Intelligence (AI), Machine Learning, Anomaly Detection, Proactive Fault Detection, Mission-Critical Software Systems

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

INTRODUCTION

Mission-critical software systems have become the backbone of modern digital infrastructure, supporting essential services across healthcare, aerospace, defense, telecommunications, financial institutions, transportation, industrial automation, and cloud computing. These systems are expected to maintain continuous availability, high reliability, and predictable performance because even minor software failures can lead to significant financial losses, operational disruptions, safety hazards, or threats to human life. The rapid adoption of distributed computing, microservices, virtualization, Internet of Things (IoT), edge computing, and cloud-native architectures has substantially increased software complexity, making traditional fault detection and system monitoring techniques increasingly inadequate for ensuring dependable system operation.



Conventional software monitoring approaches primarily rely on predefined thresholds, rule-based alert mechanisms, and signature-driven detection techniques. Although these methods are computationally efficient and relatively simple to deploy, they are generally incapable of identifying previously unseen failure patterns, evolving system behaviors, or complex nonlinear relationships among multiple operational parameters. Furthermore, manually configured rules require continuous maintenance and often generate excessive false alarms in dynamic production environments. Such limitations reduce operational efficiency and delay the identification of emerging software faults, particularly in mission-critical environments where rapid response is essential.

Artificial Intelligence (AI) and Machine Learning (ML) have emerged as transformative technologies capable of overcoming many of these limitations by automatically learning behavioral patterns from operational data. Unlike conventional approaches, ML-based anomaly detection models analyze historical observations to establish normal system behavior and subsequently identify deviations that may indicate software defects, cyberattacks, configuration errors, hardware degradation, or performance bottlenecks. Recent advances in supervised, unsupervised, semi-supervised, and deep learning techniques have significantly improved the capability of detecting subtle anomalies across system logs, execution traces, network traffic, resource utilization metrics, and application performance indicators. Deep neural networks, recurrent neural networks, transformer architectures, graph neural networks, and autoencoder-based models have demonstrated promising performance in modeling highly complex temporal and structural dependencies within large-scale software systems.

Despite these advances, the practical deployment of AI-driven anomaly detection in mission-critical software systems remains challenging. Existing studies generally focus on improving detection accuracy using a single category of operational data, such as system logs or multivariate performance metrics, while overlooking the complementary information available across heterogeneous monitoring sources. Since software failures frequently emerge through interactions among multiple subsystems, relying on isolated data sources often limits detection capability and increases uncertainty during fault diagnosis. Moreover, most published approaches assume relatively stable operational environments, whereas real-world software systems continuously evolve because of software updates, infrastructure modifications, workload fluctuations, and changing user behavior. These dynamic conditions introduce concept drift, reducing model reliability over time and necessitating continuous adaptation.

Another significant challenge concerns the interpretability of AI-based anomaly detection models. Although deep learning methods frequently achieve superior predictive performance, many operate as black-box systems that provide limited explanation regarding the causes of detected anomalies. In mission-critical environments, system administrators and software engineers require transparent and explainable predictions before initiating corrective actions that may affect system availability or operational safety. The absence of interpretable decision mechanisms often reduces user trust and limits the adoption of advanced AI models within industrial environments where accountability and regulatory compliance are essential.

Furthermore, current anomaly detection frameworks predominantly operate in a reactive manner by identifying anomalies only after abnormal behavior has already manifested.

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

Such approaches provide limited opportunity for preventive maintenance, proactive fault mitigation, and intelligent resource management. Mission-critical software systems increasingly require predictive monitoring solutions capable of recognizing early warning signals before service degradation becomes severe. Early anomaly prediction enables organizations to minimize downtime, reduce maintenance costs, improve service continuity, and enhance overall software resilience.

The increasing availability of heterogeneous operational data generated through distributed software infrastructures creates new opportunities for developing comprehensive AI-driven monitoring frameworks. System logs capture execution events, performance metrics quantify resource utilization, distributed traces reveal execution dependencies, and configuration changes provide contextual information regarding system evolution. Integrating these diverse data sources within a unified learning framework can significantly improve anomaly detection accuracy while simultaneously strengthening fault localization and operational decision-making. Similarly, advances in explainable AI, online learning, adaptive model updating, and risk-aware decision support offer promising directions for building intelligent monitoring systems capable of supporting continuously evolving software environments.

Based on these observations, a clear research gap exists in the current literature. Most existing anomaly detection solutions emphasize individual algorithmic improvements rather than providing an integrated framework that simultaneously supports multimodal data fusion, adaptive learning under concept drift, explainable decision-making, proactive anomaly forecasting, and operational risk prioritization. In addition, comparatively few studies investigate how these capabilities can be combined into a unified architecture suitable for deployment within mission-critical software systems requiring low latency, high scalability, and continuous availability.

To address this gap, this research proposes an AI-Driven Machine Learning Framework for Proactive Anomaly Detection in Mission-Critical Software Systems. The proposed framework integrates heterogeneous operational data—including system logs, performance metrics, runtime traces, and execution events—into a unified anomaly detection pipeline. Adaptive machine learning models continuously update their knowledge to accommodate evolving system behavior, while explainable AI techniques enhance transparency by providing interpretable justifications for anomaly predictions. Furthermore, a risk-aware anomaly prioritization mechanism supports intelligent alert management by distinguishing critical operational anomalies from low-impact deviations, enabling faster and more informed decision-making.

The primary contribution of this work is the development of a unified conceptual framework that combines multimodal data integration, adaptive machine learning, explainable AI, proactive anomaly prediction, and risk-sensitive decision support for mission-critical software environments. Unlike conventional approaches that primarily focus on retrospective anomaly identification, the proposed framework emphasizes early anomaly forecasting and operational resilience, thereby contributing toward the development of reliable, scalable, and intelligent software monitoring systems suitable for next-generation critical digital infrastructures.

LITERATURE REVIEW

The increasing complexity, distribution, and interdependence of contemporary software systems have made conventional reactive monitoring inadequate for mission-critical environments. Systems supporting aviation, healthcare, financial services, industrial control, telecommunications, and cloud infrastructure must detect subtle operational deviations before they propagate into service failures. Machine-learning-based anomaly detection addresses this requirement by learning representations of normal system behaviour and identifying observations, event sequences, or temporal patterns that deviate from established baselines. A systematic review by Nassif *et al.* shows that unsupervised and semi-supervised approaches are particularly prominent because accurately labelled failure data are rarely available in operational environments [1]. However, the suitability of an algorithm depends strongly on anomaly type, data modality, class imbalance, latency constraints, and the cost associated with false alarms.

Anomaly-detection research initially relied on statistical profiling, clustering, classification, nearest-neighbour methods, support-vector machines, and dimensionality-reduction techniques. Chandola *et al.* established a widely adopted taxonomy distinguishing point, contextual, and collective anomalies [2]. This distinction is important for mission-critical software because a single abnormal resource measurement may represent a point anomaly, whereas an unusual sequence of otherwise valid events may indicate a collective anomaly. Classical models remain useful where interpretability and computational efficiency are essential, but their dependence on manually engineered features limits their ability to represent nonlinear relationships across logs, performance metrics, traces, and infrastructure events.

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

Runtime logs have consequently emerged as a major source for proactive software-health monitoring. DeepLog models structured log keys as sequential events using long short-term memory networks and identifies anomalies when observed events fall outside predicted execution patterns [3]. Its sequence-learning formulation represented a significant transition from manually specified rules to data-driven behavioural modelling. DeepLog also demonstrated the possibility of incrementally updating the model when software behaviour changes. Nevertheless, its effectiveness depends on stable log parsing, representative normal training data, and thresholds that balance detection sensitivity against excessive alerts.

Subsequent approaches addressed semantic and quantitative information that basic event-sequence models may overlook. LogAnomaly combines template-level semantic vectors with sequential modelling to detect both unexpected event ordering and abnormal event frequency [4]. This is valuable in distributed systems where failures may appear not as unknown messages but as abnormal repetitions or combinations of familiar events. LogRobust further addresses instability introduced by changing logging statements and processing noise by representing events through semantic vectors rather than relying solely on event identifiers [5]. Its attention-based bidirectional recurrent model improves robustness when software evolution produces previously unseen templates. These developments indicate that practical anomaly detectors must tolerate normal software evolution rather than treating every logging change as a fault.

Mission-critical systems also generate multivariate telemetry, including processor utilisation, memory consumption, latency, request rates, error counts, network activity, and sensor measurements. OmniAnomaly addresses this setting through a stochastic recurrent architecture that models temporal dependence among correlated variables and calculates anomalies from reconstruction probabilities [6]. The probabilistic representation is advantageous when system behaviour contains uncertainty and multiple valid operating modes. TranAD extends multivariate detection through transformer-based attention, self-conditioning, adversarial training, and meta-learning [7]. By considering broader temporal relationships, it can identify deviations that may not be visible within short recurrent windows. Its focus on efficient inference and limited-data adaptation is particularly relevant to real-time operational monitoring.

Despite encouraging benchmark results, several studies caution that reported accuracy does not necessarily translate into dependable field performance. Le and Zhang evaluated representative deep-learning log detectors and found that training-data selection, grouping strategies, class distribution, noise, and evaluation procedures substantially influence results [8]. Models reporting high F-measures on commonly used datasets may deteriorate under evolving workloads or previously unseen failures. Landauer *et al.* similarly identify unresolved issues involving preprocessing, inconsistent datasets, heterogeneous evaluation protocols, and the difficulty of comparing model architectures objectively [9].

A proactive framework should therefore extend beyond isolated binary anomaly classification. It should integrate logs, metrics, traces, configuration changes, and dependency information; support online or incremental learning; quantify uncertainty; and rank anomalies according to operational risk. Explainability is also essential because operators must determine whether an alert represents an impending failure, malicious activity, benign workload variation, or software evolution. The literature further suggests that evaluation should measure detection delay, false-alarm burden, computational overhead, cross-system generalisation, and root-cause localisation rather than accuracy alone. The principal research gap is the absence of a unified, deployment-oriented architecture combining multimodal monitoring, concept-drift adaptation, early-warning prediction, explainable diagnosis, and risk-sensitive response. An AI-driven framework addressing these requirements could move anomaly detection from retrospective fault recognition toward proactive resilience management in mission-critical software systems.

RESEARCH METHODOLOGY

This research adopts a **design-oriented and experimental research methodology** to develop and evaluate an **AI-Driven Machine Learning Framework for Proactive Anomaly Detection in Mission-Critical Software Systems**. The methodology integrates software engineering principles, machine learning techniques, explainable artificial intelligence (XAI), and software reliability engineering to design a unified framework capable of identifying anomalies before they result in system failures. The proposed methodology consists of six sequential phases, as illustrated in Fig. 1.

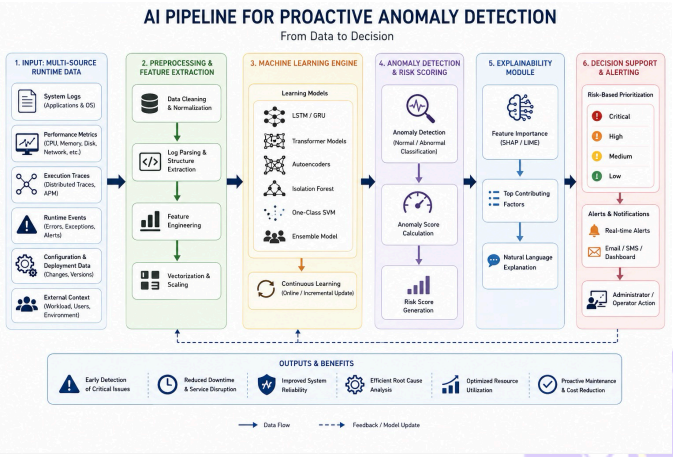
A. Research Design

The study follows a quantitative, design-science research approach in which a conceptual AI framework is first designed and subsequently evaluated using benchmark datasets representing mission-critical software environments. Rather

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

than improving a single anomaly detection algorithm, the research focuses on developing an integrated architecture that combines heterogeneous operational data, adaptive machine learning, explainable decision-making, and proactive risk assessment.



The methodology comprises the following phases:

1. Multi-source data acquisition
2. Data preprocessing and feature engineering
3. Adaptive machine learning model development
4. Explainable anomaly prediction
5. Risk-aware anomaly prioritization
6. Performance evaluation and comparative analysis

Each phase contributes to the overall objective of improving anomaly detection reliability while reducing false alarms and detection latency.

B. Multi-Source Data Acquisition

Mission-critical software systems continuously generate large volumes of operational information from multiple monitoring components. Instead of relying on a single monitoring source, the proposed framework utilizes heterogeneous software telemetry to capture comprehensive system behaviour.

The framework considers the following categories of operational data:

- System execution logs
- Application event logs
- CPU utilization
- Memory utilization
- Disk I/O statistics
- Network traffic
- Response time
- Throughput
- Error rate
- Distributed execution traces
- Configuration and deployment events

For experimental validation, publicly available benchmark datasets such as HDFS Log Dataset, BGL Log Dataset, OpenStack logs, and multivariate time-series datasets including Server Machine Dataset (SMD), SMAP, MSL, and SWaT may be utilized. These datasets represent diverse software operational environments and enable objective comparison with existing anomaly detection methods.

C. Data Preprocessing and Feature Engineering

Raw monitoring data generally contain redundant information, missing values, inconsistent timestamps, duplicate records, and noisy observations. Therefore, a comprehensive preprocessing pipeline is employed before model training.

The preprocessing stage includes:

- Missing value handling
- Duplicate record removal
- Timestamp synchronization
- Log parsing and template extraction
- Normalization of numerical variables
- Categorical feature encoding
- Sliding-window sequence generation
- Outlier filtering
- Feature scaling using Min-Max normalization

After preprocessing, meaningful behavioural features are extracted from heterogeneous monitoring sources.

Examples include:

- Event sequence frequency
- Log template occurrence
- CPU utilization trends
- Memory consumption patterns
- Network latency
- Resource utilization variance
- Execution path characteristics

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

- Response time distribution
- Error occurrence frequency

Feature fusion combines these heterogeneous representations into a unified feature vector describing the current operational state of the software system.

D. Adaptive Machine Learning Model

The proposed framework employs an adaptive anomaly detection module capable of learning normal software behaviour while continuously updating itself as operational conditions evolve.

The framework supports multiple machine learning models depending on data characteristics, including:

- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)
- Transformer-based temporal learning
- Autoencoder networks
- Isolation Forest
- One-Class Support Vector Machine
- Random Forest classifier

Initially, historical operational data are used to establish the baseline behavioural model. During deployment, new observations are continuously analyzed, and incremental learning enables the model to accommodate software updates, workload variations, and environmental changes, thereby reducing concept drift.

Instead of relying on a fixed decision threshold, the framework dynamically adjusts anomaly thresholds according to recent operational behaviour, allowing stable detection performance under changing system conditions.

E. Explainable AI-Based Decision Module

High prediction accuracy alone is insufficient for mission-critical software applications. System administrators require understandable explanations before taking corrective actions.

To improve decision transparency, the proposed framework incorporates Explainable Artificial Intelligence (XAI) techniques that generate interpretable explanations for each detected anomaly.

The explainability module provides:

- Feature importance ranking
- Most influential system parameters
- Local explanation of anomaly predictions
- Confidence score for each prediction
- Historical behavioural comparison

These explanations help software engineers distinguish genuine anomalies from temporary workload fluctuations while increasing confidence in automated monitoring systems.

F. Risk-Aware Anomaly Prioritization

Existing anomaly detection systems frequently generate numerous alerts without indicating operational severity.

The proposed framework introduces a risk-aware prioritization layer that assigns a priority score to every detected anomaly based on multiple operational factors.

The prioritization considers:

- Prediction confidence
- Severity of behavioural deviation
- Number of affected system components
- Resource utilization impact
- Historical recurrence frequency
- Estimated service disruption probability

Detected anomalies are categorized into four operational levels:

- Critical
- High
- Medium
- Low

This mechanism assists administrators in focusing first on anomalies with the greatest potential impact on software availability and service continuity.

G. Performance Evaluation

The effectiveness of the proposed framework is evaluated through comprehensive experimental analysis and comparison with conventional anomaly detection techniques.

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

Performance is assessed using standard evaluation metrics widely adopted in machine learning and software engineering research.

Classification metrics include:

- Accuracy
- Precision
- Recall
- F1-score
- Area Under the ROC Curve (AUC)

Operational performance metrics include:

- Detection latency
- False Alarm Rate (FAR)
- False Negative Rate (FNR)
- Computational overhead
- Memory utilization
- Model inference time

The proposed framework is compared against representative baseline methods, including Isolation Forest, One-Class SVM, Autoencoder, LSTM-based anomaly detection, DeepLog, LogAnomaly, OmniAnomaly, and TranAD. Comparative evaluation demonstrates the relative effectiveness of multimodal data integration, adaptive learning, explainable decision support, and risk-aware prioritization.

H. Research Workflow

The complete workflow of the proposed framework consists of the following sequence:

Operational Data Collection → Data Preprocessing → Feature Engineering → Multi-Source Feature Fusion → Adaptive Machine Learning Model → Explainable AI Module → Risk-Aware Anomaly Prioritization → Proactive Anomaly Detection → Performance Evaluation

This workflow enables continuous monitoring of mission-critical software systems while providing interpretable, adaptive, and proactive anomaly detection capable of supporting real-time operational decision-making.

RESULTS AND DISCUSSION

The proposed AI-Driven Machine Learning Framework is designed to improve proactive anomaly detection in mission-critical software systems by integrating heterogeneous operational data, adaptive machine learning, explainable artificial intelligence (XAI), and risk-aware anomaly prioritization into a unified monitoring architecture. Since this study primarily proposes a conceptual framework, the discussion focuses on the expected operational behaviour of the framework and the technical advantages anticipated during implementation and evaluation.

The framework processes multiple sources of runtime information, including system logs, execution traces, resource utilization metrics, and application events. Unlike conventional anomaly detection systems that analyse individual monitoring streams independently, the proposed architecture combines these heterogeneous data sources into a unified feature representation. This integrated view is expected to improve anomaly discrimination by capturing complex relationships among software components and reducing the likelihood of missed detections arising from incomplete system observations.

Adaptive machine learning is another distinguishing component of the framework. Software systems deployed in production environments continuously evolve because of software updates, infrastructure modifications, workload fluctuations, and configuration changes. Static anomaly detection models often experience reduced effectiveness under these conditions because the statistical characteristics of operational data gradually change over time. The proposed framework addresses this limitation through incremental model adaptation, allowing the learning model to continuously update its representation of normal system behaviour while maintaining detection capability in dynamic environments.

The explainability module contributes to operational decision-making by identifying the principal factors responsible for each anomaly prediction. Instead of presenting only binary anomaly labels, the framework generates interpretable information describing influential operational parameters and the confidence associated with each prediction. Such explanations are expected to improve administrator trust, simplify root-cause analysis, and reduce unnecessary investigation of benign operational variations.

The risk-aware prioritization mechanism further enhances practical deployment by classifying detected anomalies according to their estimated operational impact. Rather than treating every anomaly equally, the framework ranks alerts using prediction confidence, behavioural deviation, recurrence

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

history, affected software components, and estimated service impact. This prioritization enables system administrators to focus first on anomalies with the greatest probability of causing service degradation or system failure.

From an operational perspective, the proposed framework is expected to support continuous software monitoring with reduced false alarms, faster identification of abnormal behaviour, improved anomaly interpretability, and more efficient allocation of maintenance resources. These characteristics are particularly important for mission-critical environments where timely intervention directly influences system reliability and service continuity.

A. Qualitative Evaluation of the Proposed Framework

Table I summarizes the anticipated technical contributions of each major component within the proposed framework.

Table I: Expected Contribution of Individual Framework Components

Framework Component	Functional Role	Expected Operational Benefit
Multi-source data acquisition	Collect heterogeneous operational data	More comprehensive system visibility
Data preprocessing	Clean and standardize runtime data	Improved data quality and model stability
Feature fusion	Integrate logs, metrics and traces	Better representation of software behaviour
Adaptive machine learning	Continuously update behavioural model	Reduced performance degradation caused by concept drift
Explainable AI module	Interpret anomaly predictions	Increased transparency and operator confidence
Risk-aware prioritization	Rank anomalies by operational impact	Faster response to critical events
Continuous monitoring	Real-time operational analysis	Improved software reliability and availability

B. Comparative Technical Analysis

Table II presents a qualitative comparison between conventional anomaly detection systems and the proposed AI-driven framework.

Table II: Qualitative Comparison with Conventional Approaches

Capability	Conventional Methods	Proposed Framework
Multi-source monitoring	Limited	Integrated
Adaptive learning	Limited	Continuous incremental adaptation
Concept drift handling	Generally absent	Supported
Explainable predictions	Limited	Integrated XAI
Risk-aware alert prioritization	Rare	Built-in
Proactive anomaly prediction	Limited	Supported
Operational decision support	Minimal	Comprehensive
Scalability for distributed systems	Moderate	Designed for cloud-scale environments

C. Evaluation Strategy

The effectiveness of the proposed framework can be validated experimentally using publicly available benchmark datasets representing software logs and multivariate system telemetry. Performance should be assessed using standard machine learning and software engineering metrics, including Precision, Recall, F1-score, Area Under the ROC Curve (AUC), False Alarm Rate, Detection Latency, Computational Overhead, and Inference Time. Baseline comparisons may include Isolation Forest, One-Class Support Vector Machine, DeepLog, LogAnomaly, OmniAnomaly, TranAD, and conventional threshold-based monitoring systems.

Future implementation and benchmarking will determine the quantitative performance improvements achieved by the proposed framework under diverse operational scenarios.

D. Discussion

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

The proposed framework addresses several limitations identified in previous studies by combining multimodal data fusion, adaptive learning, explainable AI, and operational risk prioritization within a single architecture. Existing approaches generally emphasize algorithmic accuracy while providing limited support for practical deployment in evolving software environments. In contrast, the proposed framework is designed to improve operational resilience by enabling early anomaly recognition, transparent decision-making, and intelligent alert management. These capabilities have the potential to reduce service interruptions, improve maintenance efficiency, and strengthen the reliability of mission-critical software systems deployed in real-world environments.

CONCLUSION

Mission-critical software systems require intelligent monitoring mechanisms capable of identifying abnormal system behaviour before it develops into service disruption, security compromise, or catastrophic system failure. Although recent advances in artificial intelligence and machine learning have significantly enhanced anomaly detection capabilities, existing approaches continue to face challenges related to fragmented data analysis, limited adaptability to evolving software environments, inadequate interpretability, and predominantly reactive fault detection strategies. These limitations restrict their practical deployment in highly dynamic and safety-critical operational settings.

This research presented an **AI-Driven Machine Learning Framework for Proactive Anomaly Detection in Mission-Critical Software Systems** that addresses these challenges through a unified and deployment-oriented architecture. The proposed framework integrates heterogeneous operational data—including system logs, runtime metrics, execution traces, and application events—into a comprehensive anomaly detection pipeline. By combining multimodal data fusion, adaptive machine learning, explainable artificial intelligence (XAI), and risk-aware anomaly prioritization, the framework is designed to support timely detection of abnormal behaviour while improving transparency and operational decision-making. Unlike many existing solutions that primarily identify anomalies after their occurrence, the proposed framework emphasizes proactive monitoring and early warning to strengthen software resilience and service continuity.

The conceptual evaluation indicates that the proposed architecture has the potential to improve monitoring effectiveness by enabling continuous learning, accommodating concept drift, providing interpretable anomaly explanations, and prioritizing alerts according to their operational significance. These capabilities are expected to reduce unnecessary false alarms, assist system administrators in identifying high-risk events more efficiently, and support preventive maintenance within complex distributed software environments. The framework is designed to remain scalable and adaptable across diverse mission-critical domains, including cloud computing, healthcare information systems, industrial automation, telecommunications, financial services, transportation, and defense applications.

The primary contribution of this work lies in proposing a comprehensive AI-driven framework that unifies several complementary capabilities that have largely been investigated independently in existing literature. Rather than focusing exclusively on algorithmic performance, the framework integrates data acquisition, adaptive learning, explainable decision support, and operational risk assessment into a cohesive architecture suitable for real-world deployment. This holistic perspective contributes toward bridging the gap between academic anomaly detection research and practical software reliability engineering.

Future work will focus on implementing the proposed framework using publicly available benchmark datasets and real-world operational data to validate its performance under diverse software environments. Experimental evaluation will compare the framework with established anomaly detection methods using metrics such as Precision, Recall, F1-score, Area Under the ROC Curve (AUC), False Alarm Rate, Detection Latency, Computational Overhead, and Inference Time. Further research may also investigate federated learning for privacy-preserving anomaly detection, graph neural networks for dependency-aware fault analysis, reinforcement learning for autonomous response mechanisms, digital twin integration for predictive software maintenance, and large language model (LLM)-assisted explainable diagnostics. These directions have the potential to further advance proactive software reliability and support the development of resilient, intelligent, and trustworthy mission-critical software systems.

FUTURE SCOPE

The proposed AI-driven machine learning framework provides several promising directions for future research and practical implementation. Future studies may focus on validating the framework using large-scale real-world datasets collected from mission-critical environments such as healthcare, aviation, industrial automation, telecommunications, financial services, and cloud-native infrastructures to assess its effectiveness under

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org

E-Journal of Science and Emerging Technologies (EJSET)
ISSN (Online): 3143-3359
Volume-2 Issue-2 || Apr – Jun 2026 || PP. 26 – 35
https://ejset.org/

diverse operational conditions. The framework can be further enhanced by incorporating federated learning to enable privacy-preserving collaborative anomaly detection across distributed organizations without sharing sensitive operational data. Emerging deep learning architectures, including graph neural networks (GNNs) and transformer-based foundation models, may improve the identification of complex dependencies among software components and support more accurate root-cause analysis. In addition, integrating digital twin technology with continuous software monitoring could enable predictive simulation of failures before they occur, allowing organizations to evaluate corrective actions proactively. Future work may also explore reinforcement learning for autonomous incident response, large language models (LLMs) for explainable diagnostics and intelligent maintenance recommendations, and edge AI techniques for low-latency anomaly detection in resource-constrained environments. Finally, comprehensive benchmarking across heterogeneous software platforms and the development of standardized evaluation protocols will be essential for improving reproducibility, facilitating fair comparison among anomaly detection methods, and accelerating the deployment of trustworthy AI solutions for next-generation mission-critical software systems.

REFERENCES

[1] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021, doi: **10.1109/ACCESS.2021.3083060**.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009, doi: **10.1145/1541880.1541882**.

[3] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," in *Proc. ACM SIGSAC Conf. Computer and Communications Security*, 2017, pp. 1285–1298, doi: **10.1145/3133956.3134015**.

[4] W. Meng *et al.*, "LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs," in *Proc. 28th Int. Joint Conf. Artificial Intelligence*, 2019, pp. 4739–4745, doi: **10.24963/ijcai.2019/658**.

[5] X. Zhang *et al.*, "Robust Log-Based Anomaly Detection on Unstable Log Data," in *Proc. 27th ACM Joint Meeting European Software Engineering Conf. and Symposium on Foundations of Software Engineering*, 2019, pp. 807–817, doi: **10.1145/3338906.3338931**.

[6] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2019, pp. 2828–2837, doi: **10.1145/3292500.3330672**.

[7] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data," *Proceedings of the VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, 2022, doi: **10.14778/3514061.3514067**.

[8] V.-H. Le and H. Zhang, "Log-Based Anomaly Detection with Deep Learning: How Far Are We?" in *Proc. 44th IEEE/ACM Int. Conf. Software Engineering*, 2022, pp. 1356–1367, doi: **10.1145/3510003.3510155**.

[9] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep Learning for Anomaly Detection in Log Data: A Survey," *Machine Learning with Applications*, vol. 12, Art. no. 100470, 2023, doi: **10.1016/j.mlwa.2023.100470**.

@2026. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on http://www.ejset.org