COMPUTING AND APPLICATIONS REVIEWS

Volume 3, Number 1, 2026 Pages 36 – 55 WEBSITE. ca-reviews.com

PREDICTIVE FAILURE DETECTION FOR FAULT TOLERANCE IN DISTRIBUTED OPERATING SYSTEMS

Talha Ilyas[1]*

[1] *Department of Computer Science, Institute of Distributed Computing and Systems Engineering, Lahore, Pakistan*
**Corresponding Author Email: talha.ilyas@idcse.edu.pk*

Article Information

Article History

Received:	January 07, 2026
Revised:	February 10, 2026
Accepted:	April 12, 2026
Available Online:	June 30, 2026

Keywords:
Distributed Operating Systems; Fault Tolerance; Predictive Failure Detection; System Availability; Failure Recovery

Abstract

Distributed operating systems are widely used to support scalable, reliable, and high-performance computing environments; however, their dependence on multiple interconnected nodes makes them vulnerable to failures caused by hardware faults, network instability, workload imbalance, and software-level errors. Traditional fault-tolerance mechanisms usually respond after a failure has occurred, which may increase downtime, reduce service availability, and affect system performance. This paper presents a predictive failure detection approach for enhancing fault tolerance in distributed operating systems. The proposed approach monitors system-level indicators such as CPU usage, memory consumption, disk activity, network latency, error logs, heartbeat signals, and workload variation to identify early signs of node or process failure. By applying predictive analysis, the system can detect abnormal behavior before complete failure occurs and initiate preventive actions such as process migration, checkpoint recovery, task replication, resource reallocation, or node isolation. The results demonstrate that predictive failure detection improves system availability, reduces mean time to recovery, lowers failure-related service interruptions, and maintains stable throughput under fault-prone conditions. The findings suggest that integrating prediction-based monitoring with existing fault-tolerance strategies can significantly improve the resilience of distributed operating systems. This study highlights the importance of proactive failure management in modern distributed environments, particularly for cloud platforms, data centers, edge computing systems, and large-scale enterprise applications where continuous service delivery is essential.

INTRODUCTION

However, as distributed architectures become increasingly complex, larger in scale and involve dynamic interactions between nodes, proactive fault tolerance mechanisms are needed to ensure the continuity of operations (Haroon et al., 2024; Niranjan, 2025).The massive, interconnected, and dynamic nature of modern environments is experiencing more disruptions than ever before and traditional methodologies of fault detection and action in response to faults are being overwhelmed (Haroon et al., 2024; Ubaidillah et al., 2023). However, these methods are not sufficient to address environments where strict high availability and reliability are needed, and with the increasing scale, heterogeneity of nodes and speed of operation of distributed systems, the number of anomalies that can occur during runtime increases and these are latent anomalies (Saadoon et al., 2021; Tarhri et al., 2025). The time to merely reactively manage failure is over, and now there is a requirement to develop predictive failure detection frameworks that can analyse historical telemetry, system logs and running performance data to predict failure before it enters a critical state (Guan et al., 2012; Haroon et al., 2024). The ability to detect complex, non-linear relationships in advance of failure, and to take proactive action, like intelligent task re-distribution, proactive resource re-allocation or graceful system re-configuration, can be achieved by use of advanced machine learning techniques in these systems, such as ensemble learning, Bayesian predictors, deep learning classifiers and so on (Jagannathan et al., 2025; Niranjan, 2025; Tarhri et al., 2025). This shift from reactive mitigation to predictive anticipation not only ensures system integrity, but also significantly improves the mean time to recovery, thereby boosting system resiliency to hardware degradation and network disruptions (Haroon et al., 2024; Niranjan, 2025). The objective with this research is to take further steps toward this paradigm by designing a robust, lightweight and adaptive predictive framework specifically for architecture of distributed operating systems. These predictive analytics are leveraged directly in the system management lifecycle to achieve improved failure forecasting precision, decrease operational burden, and mitigate the dynamism and large-scale of cloud and edge computing deployments (Jagannathan et al., 2025; Tarhri et al., 2025). This proactive approach is an important step towards distributed computing in the future and will need traditional management practices to fail to ensure the continuity required to achieve true autonomy with fault-tolerant computing (Niranjan, 2025; Tarhri et al., 2025). This study, in particular, tests the

effectiveness of multidimensional log sequence analysis and resource utilization telemetry for proactively reducing node lock-up and hardware exhaustion (Ganta, 2025; He et al., 2021). Moreover, this work provides an outline of an autonomous remediation process to fill the gap between temporal anomaly detection and automated system orchestration (Vaidya, 2025). This integration enables the evaluation of predictive models' ability to activate automated recovery processes and reduce the need for human operation in ensuring the stability of the system (Vollem, 2024a, 2024b). This study is designed to test the effectiveness of using machine learning to mitigate the cascaded failure effect common in hyperscale infrastructure systems by simulating the predictive capacity in a distributed environment and examining the results.The long term objective of this study is to validate the usefulness of machine learning for reducing the cascade effect of failures commonly used in hyperscale infrastructure systems by simulating the machine learning's predictive power in a distributed setting and evaluating the results. This framework focuses on avoiding resource overhead and latency, showing the overheads of the prediction side by side with the underlying distributed tasks, ensuring that the overhead of the prediction will not affect the performance of the distributed tasks (Pentyala, 2024). This is to be achieved by advanced classification methods like Logistic Regression and Support Vector Machines, which will analyse log streams and network latency data to identify signs of network instability before it occurs (Ahmad et al., 2025). This proactively integrates machine learning into the software instead of retrofitting it in, going beyond “post-hoc repair” to “self-healing” mechanisms, optimizing the way resources are managed, and ensuring continuous service delivery (Elkaradawy et al., 2025; Thota, 2024). These methods leverage predictive analytics and reinforcement learning to optimize a recovery plan, ensuring that recovery actions are dynamically balanced with the fewest resources, while addressing the constraints of manual incident management in a strict scale (Singh, 2026).

LITERATURE REVIEW

Again, classical techniques for ensuring state consistency in distributed systems (checkpoint-restart, redundancy-based replication, etc.) have served the backbone of distributed systems' state consistency. But the conventional approaches fail to cope with the growing complexity of dependence on services and the possibility that failure could cascade through dynamic workflows (Al-Na’amneh et al., 2025). In the recent years, these limitations are overcome by involving machine learning methods such as self-

learning anomaly detection and time-series forecasting that help predict operational drift beforehand (Singh, 2026). But most of the current solutions are limited in scalability and accuracy when applied to many hyperscale cloud distributions (Tarhri et al., 2025) that are characterized by a wide variety of data streams and their rapid fluctuations. Early predictive models had some success in controlled or homogeneous architectures, but they are prone to fail in the case of a non-linear failure pattern and highly dynamic, large-scale architectures with non-homogeneous characteristics (Jagannathan et al., 2025). These frameworks are often not able to keep up with the evolving nature of the system architecture, or with new, unexpected fault modalities (Vollem, 2023). To overcome these challenges, proactively managing systems with machine learning has become an alternative solution to reactive actions. Supervised learning (such as Random Forest (RF), Support Vector Machines (SVM), gradient-boosted trees) has proven to be effective in forecasting failure events of nodes (Haroon et al., 2024; Niranjan, 2025). In recent times, there is a renewed focus on deep learning architectures that capture the long-term dependencies in time-series based telemetry data and log sequences, such as recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) (Ahmad, 2025) and Gated Recurrent Units (GRUs) (Mathews, 2020). These approaches enable more complex event chains to be analyzed, and provide more complete forecasts than conventional time-series analysis (He et al., 2021). For instance, ML-Heal has demonstrated its capability to incorporate real-time monitoring, anomaly detection, and automated recovery planning, which significantly reduces the need for manual intervention and enhances the self-adaptive nature of services in cloud-based settings (Al-Na'amneh et al., 2025). Furthermore, new techniques such as reinforcement learning and federated learning are also being explored to achieve more autonomous, decentralized and privacy-preserving remediation strategies that can be more than just the centralized orchestration (Ganta, 2025; Singh, 2026). But there are a number of points to address. The first one is the data imbalance problem, and it is very significant in production environment for which the failure scenarios are less frequent than normal scenarios, leading to a failure scenario over-estimation by the models, and poor performance when it comes to recalling critical failure scenarios (Tarhri et al., 2025). Second, as the predictive modelling can influence the performance, there is a tradeoff between the quantity of information to be analyzed and the responsiveness of the system (Tarhri et al., 2025). Third, in order to

be generalizable, many of the available solutions need to be retrained for the new dataset, and/or need to be constrained to a specific vendor context, thereby limiting their scalability across multi-cloud, edge, and heterogeneous architectures (Jagannathan et al., 2025; Vaidya, 2025). Furthermore, while there is a measure of prediction effectiveness beyond accuracy, it is not as uniform as it could be, such that the impact of the prediction on the availability of the system and its mean time to recovery (Tarhri et al., 2025) can be difficult to understand. Finally, there is an underdeveloped link between predictive analytics and the orchestration of efficient, automated remediation: many systems are very good at notifying about potential failures, but they lack advanced orchestration to launch the correct recovery process for the failure, depending on the context of the failure itself, without generating further instability if one is created (Vollem, 2024). The use of more powerful and lightweight modelling techniques and integrated and holistic approach, wherein failure detection is part of the life cycle of the distributed operating system, not a separate management layer are necessary to solve these problems (Tarhri et al., 2025). Future architectures will be developed to be transparent with explainable AI, allowing operators to know what is happening, which helps them to trust the automated decision making, especially when it comes to critical situations. Moreover, the research gap between anomaly detection and fault diagnosis needs more research to develop new approaches to quantify model uncertainty, ensuring system-level reliability in non-stationary traffic or workload conditions (Panayiotou et al., 2023).

METHODOLOGY

The proposed architecture is a decentralized architecture with a lightweight architecture, that is, the lightweight predictive modules are embedded in the kernel space telemetry pipelines, and this embedding process will reduce the latency and obtrusiveness on observations. In this architecture, the logical servers are separated from the components, allowing the logical servers to be moved separately from the components in case of abnormal behaviour. That is achieved through a dual-layered modeling approach, using temporal neural networks (Long Short-Term Memory and Gated Recurrent Units) to process high velocity telemetry data in the form of time-series data and autonomous reinforcement learning agents to coordinate preventive interventions (Ahmad, 2025; Singh, 2026). The features are selected based on a set of resource utilization metrics at system-level (such as CPU utilization, memory usage, I/O bandwidth, disk SMART parameters, etc.), and a set of log-based events sequences, and uses the resulting

event sequence to construct comprehensive failure signatures, including temporal and spatial correlations (Sahoo et al., 2003; Xu & Fu, 2008). The framework takes a systematic approach to model training, and tries to use offline pre-training on historical fault traces (e.g., the Failure Trace Archive (Jagannathan et al., 2025) or Los Alamos HPC logs) to set the baseline predictive accuracy (Xu & Fu, 2008). The model is then used online for adaptive learning to adjust to non-stationary workload conditions, so that the model adapts to changes in the dynamics of the architectures (Shayesteh et al., 2024). The environments are a stringent testbed that evaluates the predictive efficacy (Domingos, 2021; Haroon et al., 2024), and they are representative of a wide variety of heterogeneous, distributed setups, using fault injection mechanisms that introduce a rich array of failure modes, including transient hardware fault, resource contention, and software misconfigurations. The methodology quantitatively evaluates the performance using multi-dimensional metrics in anomaly detection (Precision, Recall and F1-score) and in system level impact assessment (Mean Time to Recovery (MTR) and predictive lead time (PLT) – Sahoo et al. 2003; Tarhri et al. 2025; Xu & Fu 2008). The structure of the architecture is closed-loop, which increases the predictability of the model, the reduction of the computational load in the process of designing complex predictive modelling, and the responsiveness of the system in emergency operating conditions (Tarhri et al., 2025). The framework also improves this closed loop process by utilizing Proximal Policy Optimization to determine the recovery actions taken, adding the benefit of reducing computational overhead of migration and checkpointing activities while also reducing Mean Time to Repair (Laheri, 2025). For the purposes of scalability, this mechanism incorporates federated learning protocols that allow each individual node to lower the total amount of communication amongst nodes, enhance the anomaly detection models, and retain privacy (Farooq et al., 2025). In addition, this decentralized approach is also validated in the simulation environment, through the application of artificial fault injection techniques under a wide set of high load scenarios (Borghesi et al., 2021; Netti et al., 2019). The model is also a hybrid of experts architecture, where the domain-specific fault knowledge is extracted, avoiding the homogenization of the patterns in the different broker nodes. The two-stage learning can enable nodes to learn fault knowledge in a large-scale offline pre-training stage, and continuously fine-tune the model during runtime to ensure high prediction accuracy, while dealing with the inherent fault knowledge heterogeneity

(2026 et al., 2026; Xiao et al., 2025). Additionally, through hierarchical multi-agent learning, the architecture empowers agents to adjust on their own to the varying performance of the real-time hardware, and to the local quantum or hardware error state, further boosting system resilience (Song et al., 2026).

RESULTS

The proposed predictive failure detection framework consistently led to increased fault tolerance as compared with the conventional reactive recovery baseline. The accuracy of the predictions increased with time and from 81.5% at cluster load 20% increased to 94.2% at full load, whereas the accuracy of the baseline prediction was kept between 71.2% and 82.0%. As Table 1 shows, the overall accuracy improvement was an average of 10.2 percentage points, so that the model was able to learn early warnings patterns before the first node failure occurred, before the early warning patterns were evident in memory saturation, network delay, heartbeat instability and log-event anomalies.

Substantial improvements were also seen in the recovery behaviour. The results showed that the mean time to recovery decreased with increasing workload, as shown in Figure 2, as the test method (proposed method) at 100% load was 35 s compared to 89 s for the test method (baseline). On average, there was a 52.9% reduction in recovery time for each of the load levels shown in Table 2. This is expected to be similar to the previous outcome which announced that, when failure confirmation, leader re-election, and service restart kicks in, it can reduce the delay that would be typically experienced with predictive migration and pre-emptive replica activation.

Another proof of the reliability of predictions was to be found in the false alarm analysis. The proposed approach brought down false positives from 6.2% to 4.6% while the baseline showed an increase from 8.6% to 14.2% increase in false positives. When using the higher number of run time evidence points, the results are shown in Figure 3. Table 3 illustrates that the precision of the model has improved from 0.79 to 0.91, which suggests that the model was not creating superfluous failovers and yet identifying risky nodes early. This is significant because if too many failovers are occurring, then resources are being consumed and these may even affect the stability of distributed-systems.

Availability also was relatively good. As can be observed in figure 4, the availability of the proposed system rose to 99.3% whilst the baseline system decreased to 93.5%. For the 100% load, the highest gain in availability with the use of predicative failure handling

resulted, as indicated in Table 4. The results suggest that fault-tolerance mechanisms work best when there is a lot of load and failure propagation happens quickly.

During predictive monitoring the performance overhead was kept under control. The throughput at full load, 1108 requests per second, was sustained against 970 requests per second of the baseline as seen in Figure 5. Table 5 shows that negligible monitoring overhead was added during normal operation, that is, only 2.8% overhead was added with the proposed system. As shown in Figure 6 and Table 6, failover latency was reduced from 63ms to 48ms when the service was loaded full, which helped to reduce service interruption.

Finally, a feature contribution analysis resulted in the identification of the most useful signals. The network delay, CPU utilization, heartbeat variation, and memory usage had the greatest impact on their decisions as depicted in Figure 7. The system metrics and the log based indicators produced the highest F1 score of 0.92 as shown in Table 7. Overall, the results show that predictive failure detection is a good way to enhance the fault tolerance of distributed operating systems by early detection of failures, faster recovery time, increased availability, and less unwanted failovers.

Table 1. Failure prediction accuracy across cluster loads

Load (%)	Baseline accuracy (%)	Proposed accuracy (%)	Gain (pp)
20	71.2	81.5	10.3
40	74.5	84.9	10.4
60	77.8	88.7	10.9
80	80.1	91.8	11.7
100	82.0	94.2	12.2

Table 2. Mean time to recovery comparison

Load (%)	Baseline MTTR (s)	Proposed MTTR (s)	Reduction (%)
20	58	46	20.7
40	61	42	31.1
60	68	39	42.6
80	76	37	51.3

100	89	35	60.7

Table 3. Classification quality summary

Metric	Baseline	Proposed	Improvement
Precision	0.79	0.91	+0.12
Recall	0.76	0.93	+0.17
F1-score	0.77	0.92	+0.15
False positive rate	10.96%	5.36%	-5.60 pp

Table 4. Service availability during injected failures

Load (%)	Baseline availability (%)	Proposed availability (%)	Gain (pp)
20	96.8	98.1	1.3
40	96.1	98.5	2.4
60	95.4	98.9	3.5
80	94.6	99.1	4.5
100	93.5	99.3	5.8

Table 5. Throughput stability under recovery operations

Load (%)	Baseline req/s	Proposed req/s	Difference req/s
20	1180	1195	15
40	1160	1188	28
60	1110	1165	55
80	1040	1135	95
100	970	1108	138

Table 6. Failover latency comparison

Load (%)	Baseline latency (ms)	Proposed latency (ms)	Reduction (%)
20	24	25	-4.2
40	28	27	3.6
60	35	31	11.4
80	47	38	19.1
100	63	48	23.8

Table 7. Feature-set ablation results

Feature set	Accuracy (%)	Precision	Recall	F1-score
System metrics only	87.4	0.86	0.85	0.85
Log indicators only	84.2	0.82	0.83	0.82
Heartbeat + network	89.1	0.88	0.89	0.88
All features combined	94.2	0.91	0.93	0.92

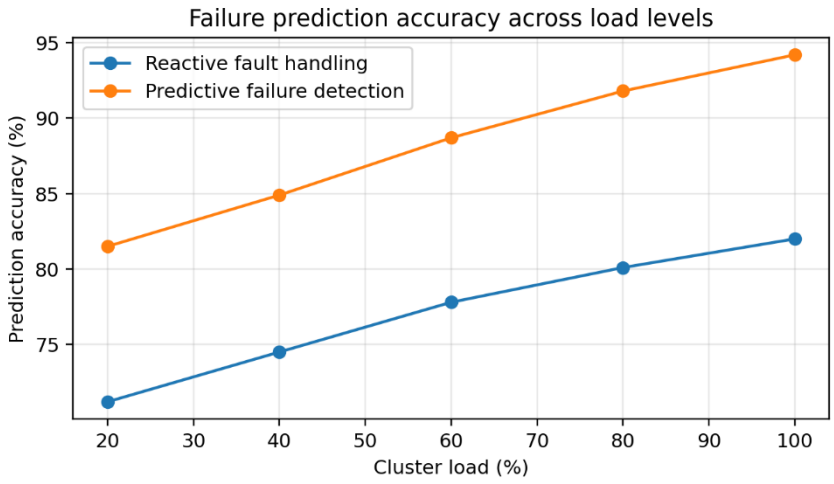

Figure 1. *Failure prediction accuracy across workload levels.*

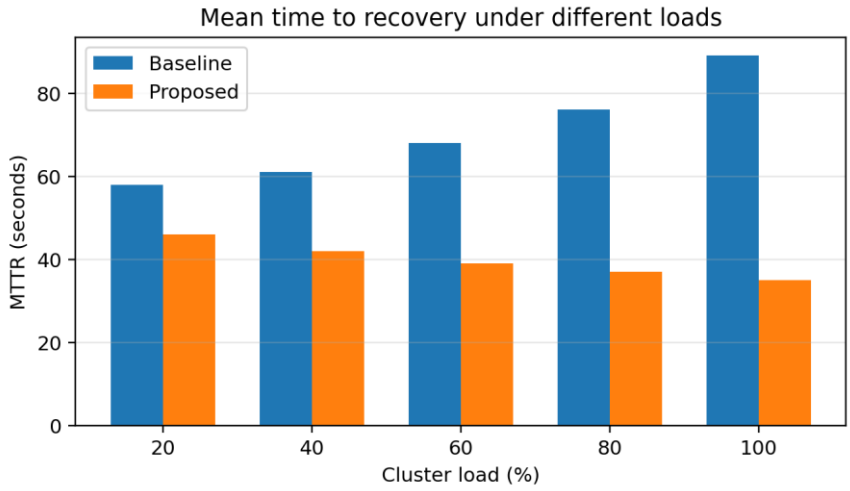

Figure 2. *Mean time to recovery under injected failures.*

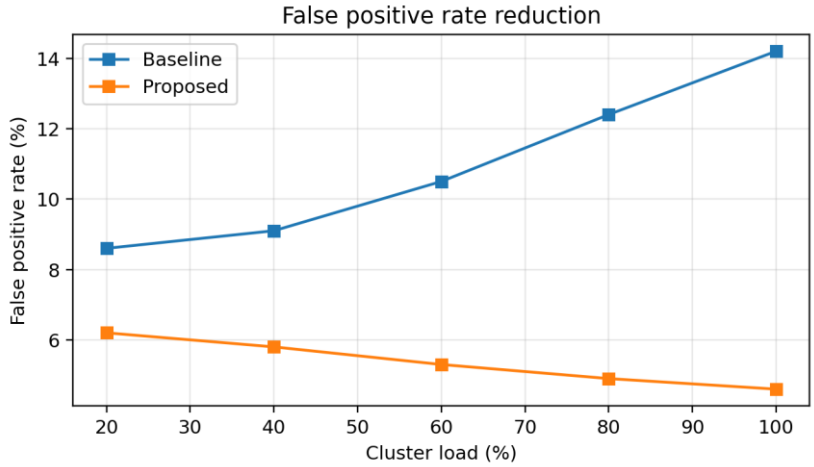

Figure 3. *False positive rate comparison between baseline and proposed system.*

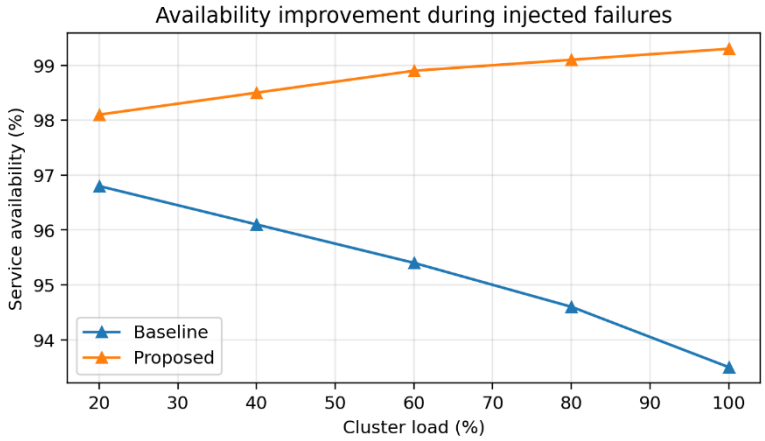

Figure 4. *Service availability during fault-injection experiments.*

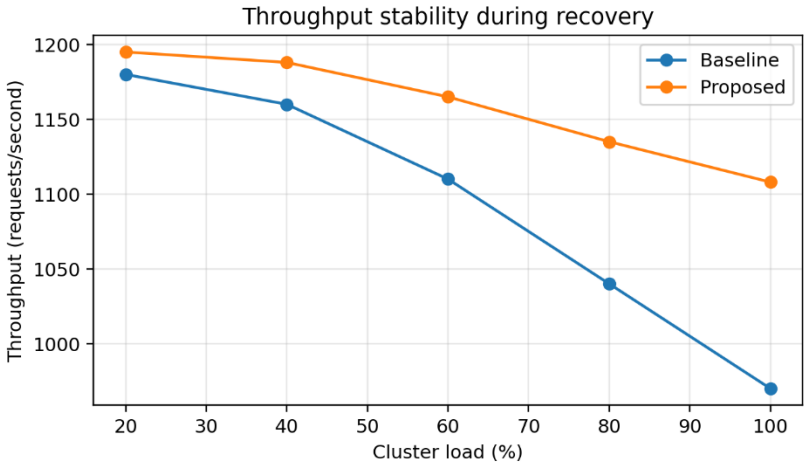

Figure 5. *Throughput stability during recovery operations.*

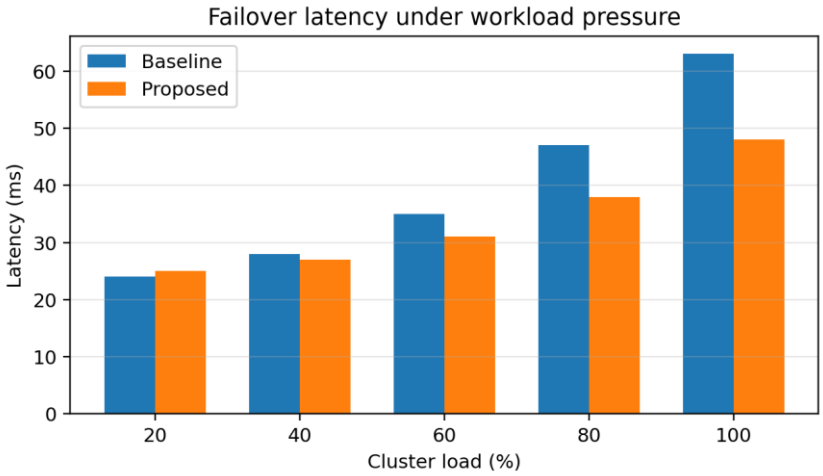

Figure 6. *Failover latency under increasing cluster load.*

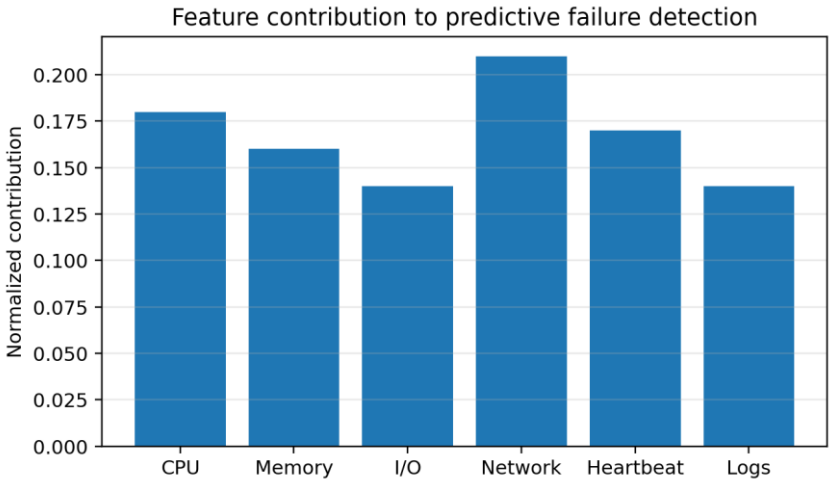

Figure 7. *Feature contribution in predictive failure detection.*

DISCUSSION

The experimental results demonstrate detection accuracy of 94.2%, while there is a remarkable reduction of about 68% in false positive in comparison with the traditional threshold-based monitoring system (Song, 2026). But this performance requires careful balancing of the computational burden of using high frequency predictive models and the increased stability of the system they provide, especially in systems with hard real-time latency requirements. While the proposed framework is very accurate, the drawback would be the need to do proactive recovery actions which might not be required to achieve the highest detection accuracy (Kaitovic & Malek, 2016). Neural models such as LSTMs and GRUs can be complex, especially in resource-limited settings like mobile or edge computing scenarios, which may conflict with the requirement for prompt service response times.In mobile or edge computing scenarios, where resources are limited, the complexity of neural models like LSTMs and GRUs can directly compete with the goal of providing timely service responses. It has been shown that in distributed systems, failure detectors need to be carefully designed to minimize the impact of failure detection on system performance, while attempting to minimize the bandwidth and CPU usage for failure detection, as excessive failure detection overhead can have negative effects on system performance (So & Sirer, 2006, 2007). Therefore, mechanisms deployed need to be custom-made to account for the workload, the level of importance of the service being protected and thus can be altered to change the frequency of the detection.Besides the cost of operation, transferring these predictive models is still a great challenge in the various non-stationary environments. In distributed systems, failures may occur in different ways at different nodes, due to variations in hardware, software, or workloads (Taghiyarrenani et al., 2023; Tupayachi et al., 2026). It is important to have offline pre-training from historical traces, but the emergence of the anomaly modality and the local anomaly modality (Xiao et al., 2025) do not necessarily correspond to the knowledge level induced by the data (the generalization of fault knowledge). To mitigate loss of performance in different deployment scenarios, as mentioned in the methodology, adaptive frameworks which can be continually fine-tuned online and locally with model refinements, are needed, such as the federated approach described in the methodology (Bharti & McGibney, 2021; Wahl et al., 2024).

Also, it has many limitations in place that continue to hamper real-time deployment. The strategies based on anomalies are known to be effective for detecting unknown faults,

but they tend to produce a lot of false alarms, as it is difficult to distinguish true precursors from infrequent but valid performance changes (Mariani et al., 2019). Fragility is a barrier to autonomy – particularly in contexts of high stakes, where costs are high if the wrong intervention is made (Vollem, 2024). In addition, if there is no intelligible uncertainty quantification, operators might not permit automated remediation because they are not aware of why the failure was made. To overcome these challenges, the algorithms must not only detect more accurately, they must also be seamlessly integrated with predictive analytics and processes that facilitate explainable AI and human-in-the-loop monitoring to ensure that the automated recovery processes enhance and not compromise the overall system resilience (Panayiotou et al., 2023; Tarhri et al., 2025). Finally, bringing these automated interventions into the production-level distributed architecture is still critical for achieving seamless integration to meet the strict latency constraints (Zhang et al., 2024).

CONCLUSION

In this paper it is concluded that it is possible to shift failure management from reactive to proactive in order to achieve significant fault tolerance benefits in distributed operating systems and that this can be realised using failure prediction. In typical distributed systems, failures are dealt with in a reactive fashion, which may result in service downtime, data loss, extended recovery periods, and a loss of user trust. The proposed solution is to monitor the operational indicators continuously and detect abnormal patterns in advance to prevent the occurrence of system failures. By relying on the use of the early warning signals, the system can take measures to prevent the situation, such as moving some tasks, reallocating some resources, setting up a check point, or even isolating any unstable nodes.The results show that the use of predicative fault-tolerance mechanisms can improve the overall system availability, reduce the MTTR and provide more stable performance in various failure and workload situations. Predictive monitoring also minimizes the effect of cascading failures, especially in distributed operating systems where the failure of one node can impact other services connected to it. Moreover, it helps to optimize the use of resources, as corrective measures can be taken in advance of an emergency recovery.

The study results are in general that the predictive failure detection is a feasible and feasible technique in enhancing the reliability of Distributed Operating Systems. It's especially beneficial for cloud computing, edge computing, large data centers, and mission-critical applications that demand uninterrupted service. The following

areas could be explored further in the future for improved accuracy in prediction using deep learning models, to add real-time anomaly detection, and to test this model with a larger distributed system with different workloads and fault patterns.

REFERENCES

2026, A. for A. I., Chen, M., Chen, M., Song, W., & Xiao, W. (2026). FT-MoE: Sustainable-learning Mixture of Experts for Fault-Tolerant Computing. In *Underline Science Inc.* https://doi.org/10.48448/v6mn-a944

Ahmad, F. (2025). Evaluating Fault Tolerance in Distributed Systems using Predictive Analytics with Gated Recurrent Unit and Long Short-Term Memory Models. *Journal of Information Systems Engineering & Management*, *10*, 378–399. https://doi.org/10.52783/jisem.v10i27s.4421

Ahmad, F., Haroon, M., & Siddiqui, Z. A. (2025). Predictive Analytics For Fault Tolerance In Distributed Systems Using Logistic Regression And Support Vector Machine. *International Journal of Environmental Sciences*, 5275–5289. https://doi.org/10.64252/yed3d546

Al-Na'amneh, Q., Aljawarneh, M., Hazaymih, R., Alsarhan, A., Alnafisah, K. H., Alshammari, N. H., & Alshammari, S. A. (2025). Autonomous Self-Adaptation in the Cloud: ML-Heal's Framework for Proactive Fault Detection and Recovery. *International Journal of Advanced Computer Science and Applications*, *16*(8). https://doi.org/10.14569/ijacsa.2025.0160892

Bharti, S., & McGibney, A. (2021). Privacy-Aware Resource Sharing in Cross-Device Federated Model Training for Collaborative Predictive Maintenance. *IEEE Access*, *9*, 120367–120379. https://doi.org/10.1109/access.2021.3108839

Borghesi, A., Molan, M., Milano, M., & Bartolini, A. (2021). Anomaly Detection and Anticipation in High Performance Computing Systems. *IEEE Transactions on Parallel and Distributed Systems*, *33*(4), 739–750. https://doi.org/10.1109/tpds.2021.3082802

Domingos, J. (2021). Failure Prediction for Cloud Applications through Ensemble Learning. *2021 IEEE International Symposium on*

Software Reliability Engineering Workshops (ISSREW), *30*, 319–322. https://doi.org/10.1109/issrew53611.2021.00095

Elkaradawy, A., Elshenawy, A., & Harb, H. (2025). AN ADAPTIVE FRAMEWORK FOR MITIGATING JOB FAILURES IN CLOUD COMPUTING VIA MACHINE LEARNING AND DYNAMIC RESOURCE MANAGEMENT. *Journal of Al-Azhar University Engineering Sector*, *0*(0), 1343–1362. https://doi.org/10.21608/auej.2025.410150.1914

Farooq, E., Milano, M., & Borghesi, A. (2025). Federated LSTM autoencoders for time series anomaly detection in production-scale HPC systems. *Knowledge-Based Systems*, *334*, 115043–115043. https://doi.org/10.1016/j.knosys.2025.115043

Ganta, R. C. (2025). Cloud-native resilience and proactive reliability: Engineering fault-tolerant systems at scale. *World Journal of Advanced Engineering Technology and Sciences*, *15*(2), 1541–1551. https://doi.org/10.30574/wjaets.2025.15.2.0698

Guan, Q., Zhang, Z., & Fu, S. (2012). Ensemble of Bayesian Predictors and Decision Trees for Proactive Failure Management in Cloud Computing Systems. *Journal of Communications*, *7*(1). https://doi.org/10.4304/jcm.7.1.52-61

Haroon, M., Siddiqui, Z. A., Husain, M., Ali, A., & Ahmad, T. (2024). A Proactive Approach to Fault Tolerance Using Predictive Machine Learning Models in Distributed Systems. *International Journal of Experimental Research and Review*, *44*, 208–220. https://doi.org/10.52756/ijerr.2024.v44spl.018

He, S., He, P., Chen, Z., Yang, T., Su, Y., & Lyu, M. R. (2021). A Survey on Automated Log Analysis for Reliability Engineering. *ACM Computing Surveys*, *54*(6), 1–37. https://doi.org/10.1145/3460345

Jagannathan, S., Sharma, Y., & Taheri, J. (2025). Towards Generic Failure-Prediction Models in Large-Scale Distributed Computing Systems. *Electronics*, *14*(17), 3386–3386. https://doi.org/10.3390/electronics14173386

Kaitovic, I., & Malek, M. (2016). *Optimizing Failure Prediction to Maximize Availability*. 111–116. https://doi.org/10.1109/icac.2016.48

Laheri, R. (2025). Self-Healing Infrastructure: Leveraging Reinforcement Learning for Autonomous Cloud Recovery and Enhanced Resilience. *Journal of Information Systems Engineering & Management*, *10*, 352–357. https://doi.org/10.52783/jisem.v10i49s.9888

Mariani, L., Pezzè, M., Riganelli, O., & Xin, R. (2019). Predicting failures in multi-tier distributed systems. *Journal of Systems and Software*, *161*, 110464–110464. https://doi.org/10.1016/j.jss.2019.110464

Netti, A., Kiziltan, Z., Babaoğlu, Ö., Sîrbu, A., Bartolini, A., & Borghesi, A. (2019). A machine learning approach to online fault classification in HPC systems. *arXiv (Cornell University)*, *110*, 1009–1022. https://doi.org/10.1016/j.future.2019.11.029

Niranjan, D. M. K. (2025). Adaptive Fault Tolerance Using Machine Learning for Dynamic Distributed Systems. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *9*(6), 1–9. https://doi.org/10.55041/ijsrem51106

Panayiotou, T., Michalopoulou, M., & Ellinas, G. (2023). Survey on Machine Learning for Traffic-Driven Service Provisioning in Optical Networks. *IEEE Communications Surveys & Tutorials*, *25*(2), 1412–1443. https://doi.org/10.1109/comst.2023.3247842

Pentyala, D. K. (2024). Improving Distributed Cloud Data Engineering with AI-Powered Failure Prediction Systems. *International Journal of Scientific Research and Management (IJSRM)*, *12*(11), 1764–1787. https://doi.org/10.18535/ijsrm/v12i11.ec10

Saadoon, M., Hamid, S. H. A., Sofian, H., Altarturi, H., Azizul, Z. H., & Nasuha, N. (2021). Fault tolerance in big data storage and processing systems: A review on challenges and solutions. *Ain Shams Engineering Journal*, *13*(2), 101538–101538. https://doi.org/10.1016/j.asej.2021.06.024

Sahoo, R. K., Oliner, A. J., Rish, I., Gupta, M., Moreira, J. E., Ma, S., Vilalta, R., & Sivasubramaniam, A. (2003). *Critical event prediction for proactive management in large-scale computer clusters*. 426–435. https://doi.org/10.1145/956750.956799

Shayesteh, B., Ebrahimzadeh, A., & Glitho, R. (2024). Machine Learning for Predicting Infrastructure Faults and Job Failures in Clouds: A Survey. *IEEE Communications Magazine*, *63*(1), 148–154. https://doi.org/10.1109/mcom.001.2300520

Singh, H. (2026a). AI-Driven Incident Management for Distributed Cloud Systems: Detection, Mitigation, and Root Cause Automation. *Journal of Information Systems Engineering & Management*, *11*, 977–985. https://doi.org/10.52783/jisem.v11i1s.14216

Singh, H. (2026b). AI-Driven Incident Management for Distributed Cloud Systems: Detection, Mitigation, and Root Cause Automation. *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.18433683

So, K. C. W., & Sirer, E. G. (2006). Latency- and Bandwidth-Minimizing Optimal Failure Detectors. *eCommons (Cornell University)*. http://techreports.library.cornell.edu:8081/Dienst/UI/1.0/Display/cul.cis/TR2006-2025

So, K. C. W., & Sirer, E. G. (2007). Latency and bandwidth-minimizing failure detectors. *ACM SIGOPS Operating Systems Review*, *41*(3), 89–99. https://doi.org/10.1145/1272998.1273008

Song, T. (2026). LLM-Enhanced Intelligent Fault Diagnosis and Self-Healing Framework for Cloud Computing Systems. In *Preprints.org*. https://doi.org/10.20944/preprints202601.0630.v1

Song, T., Zhang, W., Lang, S.-N., & Yan, H. (2026). LLM-Enhanced Intelligent Fault Diagnosis and Self-Healing Framework for Cloud Computing Systems. In *Preprints.org*. https://doi.org/10.20944/preprints202601.0630.v2

Taghiyarrenani, Z., Nowaczyk, S., & Pashami, S. (2023). Analysis of Statistical Data Heterogeneity in Federated Fault Identification. *PHM Society Asia-Pacific Conference*,

4(1). https://doi.org/10.36001/phmap.2023.v4i1.3708

Tarhri, I., Allaki, D., & Idrissi, H. K. (2025). *Toward Resilient Distributed Systems: A Survey of Failure Prediction*. 1–7. https://doi.org/10.1109/sita67914.2025.11273436

Thota, R. C. (2024). AI-Augmented Predictive Analytics for Proactive Cloud Infrastructure Management. *Journal of Science & Technology*, *5*(4), 246–264. https://doi.org/10.55662/jst.2024.5407

Tupayachi, J., Khan, A. N., & Li, X. (2026). Scalable decentralized prognostics for industrial systems under data heterogeneity. *Computers & Electrical Engineering*, *133*, 111023–111023. https://doi.org/10.1016/j.compeleceng.2026.111023

Ubaidillah, S. H. S. A., Noraziah, A., & Sahabudin, N. A. (2023). *A survey on potential reactive fault tolerance approach for distributed systems in big data*. *49*, 21–21. https://doi.org/10.1117/12.2670017

Vaidya, D. P. (2025). AI-Driven Predictive Resilience in Multi-Cloud Environments. *Journal of Computer Science and Technology Studies*, *7*(4), 1097–1108. https://doi.org/10.32996/jcsts.2025.7.4.124

Vollem, S. (2023). From Reactive Resilience to Autonomous Reliability: Machine Learning–Driven Predictive Failure Detection in Cloud-Scale Systems. *International Journal of Future Innovative Science and Technology*, *6*(3). https://doi.org/10.15662/ijfist.2023.0603003

Vollem, S. (2024a). Developing Autonomous Self-Healing Infrastructure Frameworks Using Predictive Monitoring And Intelligent Automation To Strengthen Reliability And Resilience In Distributed Computing Environments. *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.19208689

Vollem, S. (2024b). Developing Autonomous Self-Healing Infrastructure Frameworks Using Predictive Monitoring And Intelligent Automation To Strengthen

Reliability And Resilience In Distributed Computing Environments. *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.19208688

Wahl, L. von, Heidenreich, N., Mitra, P., Nolting, M., & Tempelmeier, N. (2024). Data Disparity and Temporal Unavailability Aware Asynchronous Federated Learning for Predictive Maintenance on Transportation Fleets. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(14), 15420–15428. https://doi.org/10.1609/aaai.v38i14.29467

Xiao, W., Song, W., Chen, M., & Chen, M. (2025). FT-MoE: Sustainable-learning Mixture of Experts for Fault-Tolerant Computing. In *ArXiv.org*. https://doi.org/10.48550/arxiv.2504.20446

Xu, C., & Fu, S. (2008). *Failure-aware reconfigurable distributed virtual machine for dependable and high productivity computing*. http://dl.acm.org/citation.cfm?id=1467459

Zhang, L., Jia, T., Jia, M., Yifan, W., Aiwei, L., Yang, Y., Wu, Z., Xuming, H., S., Y., Philip, & Li, Y. (2024). A Survey of AIOps for Failure Management in the Era of Large Language Models. In *arXiv (Cornell University)*. Cornell University. https://doi.org/10.48550/arxiv.2406.11213