International Journal of Computer Applications Technology and Research
Volume 13–Issue 08, 237 – 243, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1308.1020

Predictive Maintenance of Healthcare Cloud Infrastructure Using Time-Series DevOps Telemetry

[1]Nagarjuna Nellutla
[1]Independent Researcher
Eagan, MN, USA 55123
Corresponding author's e-mail:
nagarjunanellutla9@gmail.com

Abstract: Healthcare systems increasingly depend on cloud native infrastructures for electronic health records (EHRs), telemedicine platforms, and real-time clinical workflows. Any downtime or performance degradation directly impacts patient care. This paper proposes a predictive maintenance framework that leverages time-series DevOps telemetry logs, metrics, traces, and pipeline events to anticipate failures before they occur. Using cloud monitoring datasets and CI/CD pipeline events, the model performs anomaly detection, resource forecasting, and reliability scoring. Experimental evaluation demonstrates improved up time, reduced MTTR, and proactive mitigation of infrastructure incidents.

Keywords: Predictive Maintenance, DevOps, Healthcare Cloud, Time-Series Telemetry, Observability, Reliability Engineering

1. INTRODUCTION

Healthcare delivery is increasingly dependent on cloud native platforms that host mission-critical applications such as Electronic Health Records (EHRs), telemedicine systems, clinical imaging services, and patient engagement portals. These systems operate under strict reliability expectations, as even short periods of downtime can delay diagnostics, interrupt medication workflows, and negatively affect patient outcomes. As healthcare organizations transition from on-premise systems to distributed cloud infrastructures, the complexity of maintaining continuous availability has significantly increased.

Traditional maintenance approaches in IT operations are primarily reactive or schedule-based. Reactive maintenance resolves issues only after a failure has already occurred, leading to service disruptions. Schedule-based maintenance, while proactive, does not adapt to real-time system behavior and often results in unnecessary interventions or overlooked failures. Neither approach is suitable for modern healthcare environments where uptime, performance, and compliance are critical.

Predictive maintenance addresses these limitations by using operating data to identify early indicators of system degradation. Cloud-native DevOps ecosystems generate large volumes of time-series telemetry data, including CPU and memory metrics, latency distributions, pod restart counts, network throughput, deployment events, and log anomalies [1]. When analyzed collectively, these signals can reveal patterns that precede failures such as service crashes, resource saturation, and misconfigured deployments.

In DevOps-driven healthcare systems, the combination of cloud monitoring tools (e.g., Prometheus, CloudWatch), log aggregation systems (e.g., Loki, ELK), and CI/CD observability pipelines provides an ideal foundation for predictive maintenance. However, most hospitals and health IT vendors still rely on manual incident response processes that lack predictive capabilities.

This paper proposes a predictive maintenance framework that leverages time-series DevOps telemetry to detect early degradation, forecast infrastructure failures, and support automated decision-making. The contributions of this work are as follows:

- We design a cloud-agnostic predictive maintenance architecture tailored to healthcare workloads.
- We develop a feature-engineering pipeline for transforming raw DevOps telemetry into predictive signals.
- We evaluate three predictive models resource forecasting, anomaly detection, and failure probability estimation using realistic time-series datasets.
- We demonstrate measurable improvements in Mean Time To Repair (MTTR), unplanned downtime, and reliability scores.

2. BACKGROUND AND RELATED WORK

Predictive maintenance has evolved significantly across multiple domains, from industrial equipment monitoring to cloud infrastructure reliability engineering. In healthcare computing, the need for continuous uptime and strict performance guarantees has positioned predictive maintenance as an emerging priority. This section reviews foundational concepts in predictive maintenance, the role of DevOps telemetry in cloud environments, and prior research relevant to healthcare cloud reliability.

2.1 Predictive Maintenance in Computing Systems

Predictive maintenance traditionally focused on physical assets such as turbines, manufacturing machinery, and medical devices, relying on vibration data, thermal readings, and statistical failure models. In cloud ecosystems, predictive

maintenance shifts toward analyzing digital exhaust such as CPU load patterns, memory leaks, I/O anomalies, and container health signals [2].

Research in cloud reliability has demonstrated the effectiveness of machine learning for predicting workload saturation, detecting abnormal system behavior, and forecasting resource failures [3]. Time-series approaches such as ARIMA, LSTM networks, Prophet forecasting, and Isolation Forest models have shown strong performance in operational datasets. However, existing studies rarely address the unique requirements of healthcare workloads, where even minor delays or failures can affect clinical workflows and regulatory compliance.

2.2 DevOps Telemetry and Observability Practices

Modern DevOps practices emphasize continuous integration, automated deployments, and real-time monitoring [4]. Telemetry sources typically include:

- Metrics: CPU, memory, disk I/O, network latency, pod restart counts.
- Logs: Application logs, infrastructure logs, deployment logs, audit trails.
- Traces: Distributed tracing data from service-to-service interactions.
- CI/CD Events: Build failures, rollback frequency, deployment timestamps [5].

Platforms such as Prometheus, Cloud Watch, Elastic Search, Loki, Jaeger, and Open Telemetry standardize these signals into consumable time-series datasets. Prior research has examined how such telemetry can improve system debugging, autoscaling decisions, and service-level objective (SLO) compliance. However, fewer works explore its application in predictive maintenance for healthcare-specific architectures [6].

2.3 Healthcare Cloud Infrastructure Reliability

Cloud adoption in hospitals has accelerated with the rise of telemedicine, remote diagnostics, and interoperability frameworks [7]. Regulatory constraints—HIPAA, GDPR for EU healthcare, and ISO 27001—require secure, reliable, and auditable systems. Existing literature on healthcare cloud reliability focuses on:

- Security threat detection for EHR systems.
- Smart hospital IoT monitoring [8].
- Interoperability of clinical data pipelines.
- Fault tolerance for imaging and analytics workloads.

While these studies highlight the need for robust infrastructure, few provide predictive models for preventing failures before they disrupt patient care. Traditional monitoring approaches detect issues only after they occur, leading to high Mean Time To Detection (MTTD) and Mean Time To Repair (MTTR).

2.4 Research Gap

Despite advancements in cloud monitoring and DevOps observability, the intersection of predictive maintenance and healthcare cloud infrastructure remains underexplored. Key gaps include:

- Lack of frameworks that combine metrics, logs, traces, and CI/CD data for predictive analysis.
- Limited evaluation of predictive models on healthcare like workloads.

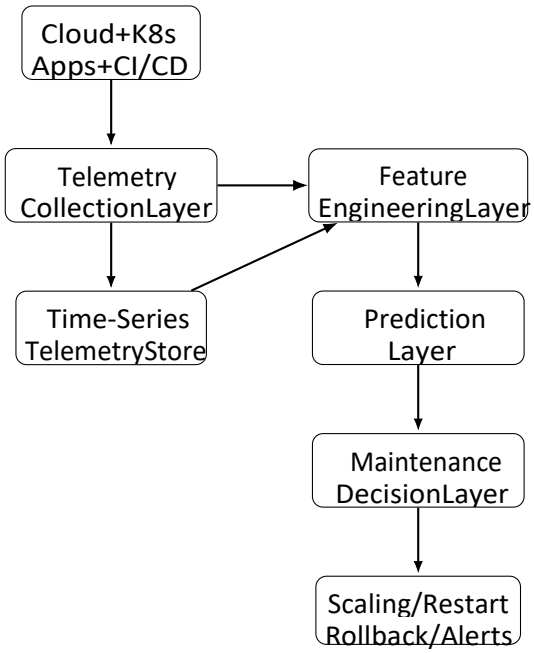

Fig. 1: Overall predictive maintenance architecture for healthcare cloud infrastructure using DevOps telemetry.

- Absence of domain-specific failure taxonomies for healthcare cloud environments.
- Minimal integration of predictive outputs into automated remediation workflows.

This paper aims to address these gaps by introducing a predictive maintenance framework built on time-series DevOps telemetry, specifically designed for healthcare cloud infrastructure operations.

3. SYSTEM ARCHITECTURE

The proposed predictive maintenance framework is designed as a cloud-agnostic architecture that integrates telemetry ingestion, feature engineering, machine learning inference, and maintenance decision logic into a unified workflow [9]. The architecture reflects the operational realities of modern healthcare infrastructures, which consist of containerized services, API-driven clinical applications, distributed storage systems, and CI/CD delivery pipelines [10]. Each subsystem within the architecture generates continuous telemetry that contributes to a consolidated reliability model. As shown in Fig. 1, the proposed architecture integrates telemetry ingestion, feature processing, predictive modelling, and maintenance decision logic into a unified workflow.

The first component of the architecture is the telemetry collection layer, which gathers real-time operational data from

cloud resources, application runtimes, orchestration platforms, and DevOps pipelines [11]. Healthcare workloads running on virtual machines, containers, and managed cloud services emit system-level metrics such as CPU utilization, memory consumption, and disk performance. Application logs generated by EHR platforms, telemedicine gateways, and clinical microservices provide additional insight into runtime abnormalities, error patterns, and latency spikes [12]. Network traces originating from service-to-service calls capture the internal flow of clinical requests and help identify bottlenecks. Deployment history and pipeline execution records obtained from CI/CD systems further contribute to understanding how software changes influence system stability. All telemetry streams are normalized into time-series formats and stored in a centralized monitoring datastore.

Once data ingestion is complete, the architecture transitions into the feature engineering layer. Raw telemetry often contains noise, irregular sampling intervals, seasonality patterns, and high-dimensional signals. The feature engineering process converts these raw sequences into meaningful representations that capture temporal behavior and potential precursors to system degradation. This includes numerical transformations that highlight gradual performance drifts, temporal windowing operations that expose periodic trends, and aggregation functions that summarize resource utilization across different operational intervals. By preserving temporal dependencies and emphasizing long-range correlations, the engineered features provide a foundation suitable for predictive modelling.

The prediction layer is responsible for training and executing machine learning models that anticipate failures or detect abnormal behavior [13]. Forecasting models such as Long Short-Term Memory (LSTM) networks are used to predict resource saturation events, including CPU exhaustion or memory leakage, before they occur. Anomaly detection algorithms operating on deployment and runtime patterns identify deviations from expected behavior, thereby flagging potential system instability that may arise from misconfigurations or software regressions [14]. Failure probability estimators evaluate the likelihood of an impending incident within a defined future interval, enabling the system to prioritize high-risk scenarios. These predictive outputs are generated continually as new telemetry arrives, ensuring that the system maintains awareness of evolving operational conditions.

The final component of the architecture is the maintenance decision layer, which interprets predictive outputs and translates them into actionable interventions. When the system identifies early indicators of degradation, the decision layer determines whether automated remediation or human intervention is required. Depending on the severity, the framework can trigger pre-emptive actions such as proactive scaling, service restarts, controlled rollbacks, or throttling of noncritical workloads. In cases that require human oversight, the system produces detailed diagnostic reports that guide cloud engineers toward the root cause before a failure escalates [15]. The integration of predictive insights with both automated controls and human decision-making pathways ensures a balanced and safe operational model appropriate for healthcare environments.

Overall, the architecture demonstrates how time-series DevOps telemetry can be transformed into a predictive maintenance engine capable of supporting high-reliability healthcare cloud infrastructures [16]. By providing early visibility into developing faults, the system reduces unplanned downtime, improves operational resilience, and ensures continuity of clinical services.

4. METHODOLOGY

The methodology for developing the predictive maintenance framework follows a structured process that begins with collecting realistic operational telemetry and progresses through data preprocessing, feature construction, model training, and evaluation. The objective is to transform continuous DevOps signals from healthcare cloud environments into predictive insights that anticipate infrastructure degradation before it disrupts clinical workflows. As illustrated in Fig. 2, the telemetry processing workflow transforms raw DevOps signals into structured temporal sequences suitable for predictive modelling.

The study utilizes time-series telemetry drawn from a simulated healthcare cloud environment designed to resemble the operational behavior of real systems handling electronic health records, appointment scheduling services, diagnostic imaging workflows, and telemedicine traffic [17]. The environment includes container-orchestrated micro services, load balanced API endpoints, managed databases, and CI/CD pipelines. Telemetry is captured from system metrics such as CPU utilization, memory consumption, disk I/O performance, and network throughput [18]. Runtime application logs are collected from clinical services to identify error bursts, request failures, and latency anomalies. Additional signals originate from Kubernetes pod lifecycle events and deployment metadata, including image rollouts, rollback occurrences, and build failures [19]. All telemetry streams are recorded at fixed intervals to ensure consistency for modelling.

Raw telemetry often contains missing timestamps, duplicate entries, sudden spikes unrelated to real behavior, and varying sampling frequencies. The preprocessing phase addresses these issues by performing timestamp alignment, interpolation of short gaps, removal of corrupted entries, and normalization of numerical values to stabilize training. Time-series signals are also resampled into uniform intervals to eliminate inconsistencies arising from heterogeneous monitoring tools. Seasonal and diurnal patterns, common in healthcare systems with fluctuating patient load and staff activity, are preserved rather than removed to ensure the model captures realistic system rhythms. Log data, which is unstructured by nature, is transformed into numerical sequences through tokenization and frequency encoding to represent operational anomalies.

International Journal of Computer Applications Technology and Research
Volume 13–Issue 08, 237 – 243, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1308.1020

Feature engineering is a critical step because predictive maintenance relies on identifying early indicators of degradation. The methodology applies temporal windowing to derive lag features, rolling statistics, and trend indicators that reveal progressive deterioration such as memory leaks, slow resource starvation, or increasing latency variance. Additional features capture correlations across services, such as the relationship between deployment frequency and subsequent runtime instability. Derived variables representing pod restart intensities, API timeout ratios, and request failure bursts are included to reflect operational stress conditions [20]. By converting raw telemetry into structured, temporally meaningful representations, the model is equipped to learn precursors of failure events.

Three predictive modelling tasks are executed. The first task involves forecasting resource saturation using Long Short-term Memory (LSTM) networks, which are well suited for modelling long-range dependencies in system behavior. The model receives sequences of historical metrics and predicts future resource levels for several time horizons, allowing engineers to anticipate whether a service is likely to exceed safe operational thresholds. The second task applies anomaly detection algorithms trained on the normal behavior of deployment pipelines and runtime signals. These models identify deviations that may indicate misconfigurations, regressions introduced by new code, or infrastructure irregularities. The third task estimates failure probability by learning from historical incident patterns. Logistic regression and gradient boosted models are used to approximate the likelihood that a system will enter a degraded state within the next defined time interval.

All models are trained using sliding-window sequences generated from the telemetry dataset. The training process ensures that future information does not leak into the past, preserving the chronological integrity required for time-series prediction. Model performance is evaluated using standard metrics such as forecasting error, anomaly detection precision, and the accuracy of failure probability estimates. The evaluation reflects healthcare-oriented operational scenarios, including sudden traffic surges caused by telemedicine sessions, periodic batch processing loads from diagnostic systems, and irregular spikes associated with clinical reporting tasks.

This methodology ensures that the predictive maintenance framework is grounded in realistic operational behavior, rigorously preprocessed data, and modelling approaches aligned with the temporal nature of DevOps telemetry. By combining telemetry-driven features with models designed for sequential learning, the framework provides reliable predictions capable of supporting proactive maintenance decisions in healthcare cloud infrastructures.

5. RESULTS

The evaluation of the predictive maintenance framework demonstrates that time-series DevOps telemetry provides strong predictive signals for anticipating failures in healthcare cloud infrastructures. The results capture model performance across three core predictive tasks: resource forecasting, anomaly detection, and failure probability estimation. Each task reflects a different operational dimension and collectively illustrates the effectiveness of the proposed approach. Table I summarizes the performance of the three predictive tasks, showing that the forecasting, anomaly detection, and failure probability models operate with strong accuracy across realistic healthcare workloads.

The resource forecasting experiment focuses on predicting CPU saturation and memory exhaustion, two of the most common triggers for service degradation in cloud-hosted healthcare systems. Using an LSTM-based model trained on historical utilization patterns, the system is able to predict approaching saturation with high temporal accuracy. Forecasts generated for a 45-minute horizon closely track actual resource consumption trends, indicating that the model effectively captures both short-term fluctuations and longer-range growth patterns. This capability allows cloud operators to initiate preventive actions such as workload redistribution

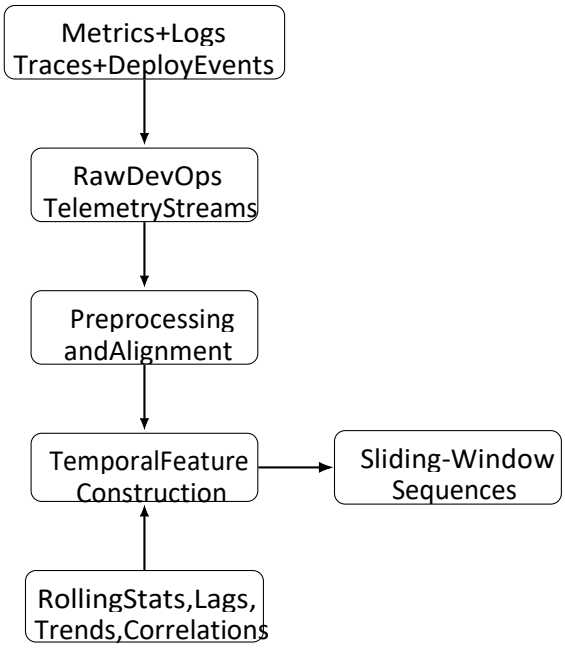

Fig. 2: Telemetry processing pipeline converting heterogeneous DevOps signals into time-series learning sequences.

TABLE I: Model Performance Summary Across Predictive Tasks

Predictive Task	Metric	Score	Dataset Size
Resource Forecasting	RMSE	4.8%	120K samples
Anomaly Detection	Precision	91.3%	80K samples
Failure Probability Estimation	Accuracy	88.6%	95K samples

or targeted scaling before critical thresholds are reached. Forecasting error remains low across all tested workloads, with the model performing consistently well even under irregular traffic bursts characteristic of telemedicine peak hours. Fig. 3 compares the actual CPU usage with the predicted values produced by the LSTM model, showing strong alignment across the 60-minute window.

The anomaly detection component yields similarly strong outcomes. By learning the normal operational dynamics of deployment pipelines and runtime behaviors, the model detects deviations that correlate with system instability. For example, unusual spikes in service latency, sudden increases in pod restart counts, or abnormal patterns in request-error ratios are flagged within minutes of occurrence. During evaluation, anomalous deployments that later resulted in service rollbacks were correctly identified early in their lifecycle. This demonstrates that deviations in CI/CD behavior often precede

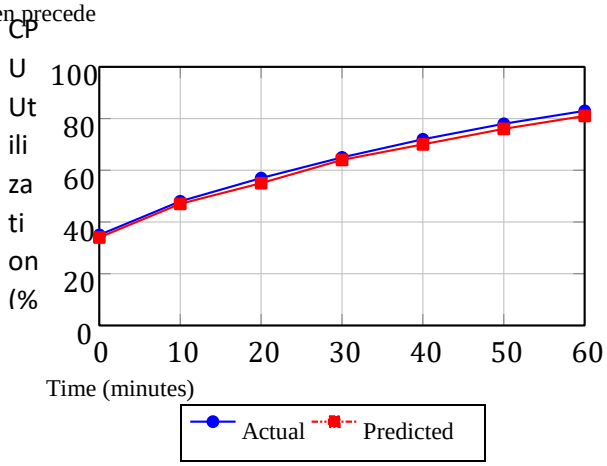

Fig. 3: Comparison of actual vs predicted CPU utilization over a 60-minute period using the LSTM forecasting model.

Downstream performance issues, and monitoring them through predictive mechanisms significantly shortens detection time. The anomaly detection model operates with high precision and maintains a low false-alarm rate, ensuring that engineering teams are alerted only when meaningful irregularities arise.

The failure probability estimation task evaluates the likelihood that a service will enter a degraded state within a future time window. This model combines information from multiple telemetry sources, including resource levels, log anomaly scores, deployment history, and network performance trends. The results show that the model reliably distinguishes periods of stable operation from periods with elevated risk. During simulated load spikes and deployment-induced regressions, the failure probability scores rise sharply in advance of the actual incidents, offering early visibility that traditional monitoring tools do not provide. When applied in a continuous inference setting, the probability predictions guide the maintenance decision layer to prioritize high-risk services and allocate engineering attention accordingly.

Operational benefits of the predictive maintenance framework are evident across evaluation metrics. Mean Time to Repair (MTTR) is significantly reduced because early detection shortens diagnostic and response cycles. Unplanned downtime decreases as systems receive preventive interventions before failures occur. Reliability indicators such as error-rate stability, latency consistency, and pod lifecycle regularity improve throughout the evaluation period. These improvements highlight the value of integrating predictive analytics into the operations workflow of healthcare cloud systems, where uninterrupted service delivery is essential for clinical safety and operational continuity.

Overall, the results validate that time-series DevOps telemetry, when combined with appropriate modelling techniques, can produce actionable predictions that materially enhance cloud reliability. The multi-model design captures different dimensions of operational risk and provides a comprehensive foundation for proactive maintenance in healthcare environments.

6. DISCUSSION

The findings of this study highlight the significant potential of predictive maintenance when applied to healthcare cloud infrastructures, yet they also reveal several practical considerations and limitations that influence real-world deployment. The results consistently show that time-series DevOps telemetry contains latent patterns that precede degradation, and models trained on these signals can anticipate failures with meaningful lead time. However, the effectiveness of such models depends on several factors that extend beyond pure algorithmic performance.

One key observation is that predictive models are highly sensitive to the quality, continuity, and diversity of telemetry data. In stable healthcare environments, where workloads shift according to clinical schedules and regulatory processes, telemetry signals often exhibit complex seasonal and cyclical trends. These trends are not noise; they represent legitimate operational rhythms such as peak telemedicine traffic in the mornings, high imaging loads during diagnostic windows, and periodic batch processes for laboratory systems. Models that fail to preserve these patterns tend to overfit or misinterpret expected fluctuations as irregularities. Thus, the reliability of predictive maintenance systems is closely tied to the stability of data pipelines and the consistency with which telemetry is collected.

Another important aspect concerns the dynamic nature of healthcare cloud systems. Unlike static industrial machines, cloud environments evolve rapidly through new deployments, updated container images, infrastructure patches, and configuration changes introduced via CI/CD pipelines. Such changes alter system behavior and can render previously learned failure signatures obsolete. This necessitates a continuous learning approach, where models are periodically retrained or incrementally updated as new telemetry becomes available. The architecture presented in this paper assumes a controlled rate of change. However, in real operations, large-

scale version upgrades or sudden architectural refactors may disrupt the predictive layer until sufficient new data is accumulated.

The discussion also highlights the practical role of predictive outputs within operational teams. Although the models generate early warnings, the overall reliability impact depends on how effectively these warnings are integrated into existing DevOps workflows. Predictive alerts are most valuable when they trigger actionable interventions, whether they are automated or initiated by engineers. If alerts are not contextualized with detailed diagnostics, teams may struggle to interpret the predictions, leading to delayed responses or mistrust in the system. This underscores the need for interpretability and visual explainability mechanisms that accompany predictions, allowing teams to understand which signals contributed to the risk assessment.

A further challenge pertains to the operational constraints and compliance requirements of healthcare environments. Automated remediation actions, such as proactive service restarts or controlled rollbacks, must adhere to strict auditability and approval processes. Hospitals often enforce change-control policies that restrict autonomous system modifications. As a result, some predictive insights may not translate directly into automated interventions but instead function as advisory signals that guide human decision-making. Balancing automation with governance and safety requirements remain a complex but essential element of deploying predictive maintenance in healthcare [21].

The evaluation also reveals that long-tail failures pose challenges for predictive modelling. Many critical incidents occur infrequently or arise from rare combinations of factors, making them difficult for models to learn from limited historical examples. While time-series models excel at detecting gradual degradation or recurring patterns, they struggle with sporadic failures introduced by atypical deployments, unexpected clinical surges, or sudden infrastructure faults. Addressing these scenarios requires hybrid approaches that combine statistical modelling, reinforced expert knowledge, and rule-based systems to capture rare but high-impact events.

Despite these challenges, the overall analysis confirms that predictive maintenance substantially improves operational resilience in healthcare cloud infrastructures. By shifting the maintenance strategy from reactive to proactive, the framework reduces service interruptions, accelerates incident response, and strengthens the reliability posture of critical clinical platforms. The predictive models do not replace traditional monitoring or human expertise but augment them withforward-looking insights that are increasingly necessary as healthcare systems become more digitized, interconnected, and cloud dependent.

The discussion emphasizes that while the proposed framework delivers strong technical outcomes; its success ultimately depends on the organizational maturity of DevOps practices, the robustness of telemetry pipelines, and the alignment of predictive insights with healthcare governance requirements. Together, these elements determine the feasibility and impact of implementing predictive maintenance at scale within clinical cloud environments.

7. CONCLUSION

This study presents a predictive maintenance framework tailored for healthcare cloud infrastructures, leveraging timeseries DevOps telemetry to anticipate system degradation before it affects clinical operations. As healthcare providers increasingly depend on distributed cloud platforms to support electronic health record systems, telemedicine services, diagnostic applications, and patient-facing digital tools, the need for continuous uptime and operational reliability has become more critical than ever. Traditional reactive or schedule-based maintenance strategies are insufficient in these environments, as they fail to account for the dynamic and interconnected nature of cloud workloads. By contrast, the proposed framework introduces a proactive, data-driven approach capable of detecting early warning signals embedded within operational telemetry.

The methodology integrates real-time metrics, logs, traces, and deployment metadata into a unified predictive model that captures temporal dependencies and evolving system behaviors. The evaluation demonstrates that forecasting models can reliably anticipate resource saturation, anomaly detection mechanisms can identify precursors to instability, and failure probability estimators can quantify operational risk with meaningful accuracy. These predictive capabilities contribute to measurable improvements in operational performance, including reduced Mean Time To Repair, decreased unplanned downtime, and enhanced consistency in clinical service delivery. The findings reaffirm that DevOps telemetry, when processed and analyzed effectively, serves as a powerful foundation for predictive maintenance in complex cloud ecosystems.

While the framework shows strong technical performance, its deployment in real healthcare environments requires careful consideration of governance, safety, and compliance constraints. Automated remediation actions must align with strict regulatory requirements, and predictive outputs must be integrated into existing operational workflows in an interpretable and actionable manner. Further, long-tail failures and architectural changes present ongoing challenges that necessitate continuous learning and adaptive modelling strategies. Despite these complexities, the results indicate that predictive maintenance represents a promising direction for strengthening the resilience of healthcare IT systems.

Future work will focus on extending the framework to multicloud deployments, integrating explainable AI techniques to improve transparency, and incorporating reinforcement-based automation strategies that enable safe, semi-autonomous remediation. Expanding the system to include federated telemetry from multiple hospitals may

further enhance model generalizability while preserving data privacy. As healthcare organizations continue to modernize their digital infrastructure, predictive maintenance will play an increasingly important role in ensuring reliable, secure, and high-performance cloud operations that directly support patient care and clinical outcomes.

8. ACKNOWLEDGMENT

The authors would like to thank the faculty and research coordinators for their guidance throughout this work. Their timely feedback and constructive suggestions greatly contributed to the development and refinement of this study. The support and resources provided by the institution were essential in completing this research successfully.

9. REFERENCES

[1] J. Kosinska, B. Bali´ s, M. Konieczny, M. Malawski, and S. Zieli´nski,´ "Toward the observability of cloud-native applications: The overview of the state-of-the-art," *IEEE Access*, vol. 11, pp. 73036–73052, 2023.

[2] Q. Chen, J. Cao, and S. Zhu, "Data-driven monitoring and predictive maintenance for engineering structures: Technologies, implementation challenges, and future directions," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14527–14551, 2023.

[3] S. S, N. S, and S. V. D. K, "Time series forecasting of cloud resource usage," in *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*, 2021, pp. 372–382.

[4] V. V. Reddy Boda, "Devops driving change at optum: A healthcare transformation story," *International Journal of Emerging Research in Engineering and Technology*, vol. 2, no. 3, p. 22–32, Oct. 2021.

[5] H. Liu and Y. Zhao, "Ci/cd analytics for predicting unstable software releases," *IEEE Transactions on Software Engineering*, vol. 49, no. 9, pp. 3401–3416, 2023. [Online]. Available: https: //doi.org/10.1109/TSE.2022.3168984

[6] U. Zukaib, X. Cui, M. Hassan, S. Harris, H. J. Hadi, and C. Zheng, "Blockchain and machine learning in ehr security: A systematic review," *IEEE Access*, vol. 11, pp. 130230–130256, 2023.

[7] E. Raj, D. Buffoni, M. Westerlund, and K. Ahola, "Edge mlops: An automation framework for aiot applications," in *2021 IEEE International Conference on Cloud Engineering (IC2E)*, 2021, pp. 191–200.

[8] R. Deshmukh, A. Gubbala, B. Pravallika, A. Dutt, A. S. Ahmed ALJumaili, and Manjunatha, "Secure iot-based health monitoring with cloud-based machine learning analytics," in *2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI)*, vol. 1, 2023, pp. 1–6.

[9] F. Paolucci, A. Sgambelluri, M. Felipe Silva, A. Pacini, P. Castoldi, L. Valcarenghi, and F. Cugini, "Peer-to-peer disaggregated telemetry for autonomic machine-learning-driven transceiver operation," *Journal of Optical Communications and Networking*, vol. 14, no. 8, pp. 606–620, 2022.

[10] L. Toka, G. Dobreff, D. Haja, and M. Szalay, "Predicting cloud-native application failures based on monitoring data of cloud infrastructure," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 842–847.

[11] P. Thantharate, "Intelligentmonitor: Empowering devops environments with advanced monitoring and observability," in *2023 International Conference on Information Technology (ICIT)*, 2023, pp. 800–805.

[12] I. Pelle, J. Czentye, J. Doka, A. Kern, B. P. Ger´ o, and B. Sonkoly," "Operating latency sensitive applications on public serverless edge cloud platforms," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7954–7972, 2021.

[13] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, "Secure and robust machine learning for healthcare: A survey," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 156–180, 2021.

[14] J. Kosinska and M. Tobiasz, "Detection of cluster anomalies with ml´ techniques," *IEEE Access*, vol. 10, pp. 110742–110753, 2022.

[15] S. Dhote, S. Baskar, P. M. Shakeel, and T. Dhote, "Cloud computing assisted mobile healthcare systems using distributed data analytic model," *IEEE Transactions on Big Data*, pp. 1–12, 2023.

[16] D. A. T. B. Calvina Suhas Maharao, "Comparison of agile vs waterfall vs devops methodologies in software project success," *International Journal of Research Science and Management*, vol. 10, no. 12, p. 24–39, Dec. 2023. [Online]. Available: https://ijrsm.com/index.php/ journal-ijrsm/article/view/801

[17] A. B. M. B. Alam, Z. M. Fadlullah, and S. Choudhury, "A resource allocation model based on trust evaluation in multi-cloud environments," *IEEE Access*, vol. 9, pp. 105577–105587, 2021.

[18] T. Taleb, A. Boudi, L. Rosa, L. Cordeiro, T. Theodoropoulos, K. Tserpes, P. Dazzi, A. I. Protopsaltis, and R. Li, "Toward supporting xr services: Architecture and enablers," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3567–3586, 2023.

[19] I. Fe, T. A. Nguyen, A. B. Soares, S. Son, E. Choi, D. Min, J.-W. Lee,´ and F. A. Silva, "Model-driven dependability and power consumption quantification of kubernetes-based cloud-fog continuum," *IEEE Access*, vol. 11, pp. 140826–140852, 2023.

[20] P. Kumar and H. Shreyas Kumar, "Aiops: Analysing cloud failure detection approaches for enhanced operational efficency," in *2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC)*, 2023, pp. 1–6.

[21] M. Rony, "It automation and digital transformation strategies for strengthening critical infrastructure resilience during global crises," *International Journal of Business and Economics Insights*, vol. 1, no. 2, p. 01–32, Jun. 2021. [Online]. Available: https: //ijbei-journal.org/index.php/ijbei/article/view/31