International Journal of application or Innovation in engineering and management

Ijaiem.com/Aug 2023/ Volume 12/Issue 2/Article No-1/64-74

ISSN: 2319-4847

Resilient ML Pipelines Using MLOps with SRE Principles and Chaos Testing for Fault-Tolerant AI Infrastructure

[1]Pramod Begur Nagaraj
Independent Researcher
Texas, USA
pbegurnagaraj@gmail.com

[2]Winner Pulakhandam
Personify Inc,Texas,USA
wpulakhandam.rnd@gmail.com

[3]Visrutatma Rao Vallu
Spectrosys,
Woburn, Massachusetts, USA
visrutatmaraovallu@gmail.com

[4]Archana Chaluvadi
Massachusetts Mutual Life Insurance Company,
Massachusetts, USA
chaluvadiarchana07@gmail.com

[5]R Padmavathy
Anna University, Coimbatore
dr.padmabarathi@gmail.com"

Article Info

Received: 31-07-2023 ***Revised: 07 -08-2023*** ***Accepted: 19-08-2023*** ***Published:30/08/2023***

Abstract

Machine learning (ML) has brought a great revolution in the industries enabling the data-based decisions and predictive analysis, yet the deployment and maintenance of ML systems with high reliability and low downtime still pose a very complex challenge. Although MLOps enhances model deployment, monitoring, and governance, it does not have resilience and fault tolerance as a core part of its process. This is the concept-based design of a fault-tolerant ML pipeline that incorporates Site Reliability Engineering (SRE) principles from MLOps along with Chaos Testing to address issues like model drift, differences in data, or adversarial attacks. The proposed design would accept automated deployments, proactive monitoring, and failure simulations to ensure minimal downtime and the best performance under sudden conditions by improving resilience to AI systems. Chaos Testing tests the pipeline by simulating the failures, while SRE takes care of proactive monitoring, fault prediction, and automated recovery. Furthermore, it predicts machine failures in predictive maintenance applications through a CNN-LSTM hybrid model. The hybrid model outperformed other simple models based on RF and SVM by an accuracy of 97.2, precision of 96.8, and an F1 score of 96.1. Additionally, it cuts Failure Recovery time (FRT) by 45% and increases Service Availability to 99.1%, while Scalability Efficiency is improved by 30%. This brings about a resilient and scalable ML infrastructure capable of continuous optimization and self-healing relevant to mission-critical applications in autonomous driving and predictive maintenance among other industrial sectors.

International Journal of application or Innovation in engineering and management

Ijaiem.com/Aug 2023/ Volume 12/Issue 2/Article No-1/64-74

ISSN: 2319-4847

Keywords: MLOps, Site Reliability Engineering, Chaos Testing, Predictive Maintenance, Resilient ML Pipeline

1. Introduction

Machine learning has brought a significant transformation in various industries, enabling them to automate processes, make data-driven decisions, and conduct predictive analysis with unprecedented accuracy [1]. Its applications span from healthcare diagnostics to financial fraud detection, where ML models provide real-time, accurate insights critical to operational success [2]. In healthcare, for example, ML algorithms can analyze medical images to assist doctors in early disease detection, while in finance, they help identify suspicious activities to prevent fraud [3]. These capabilities have revolutionized industries by improving efficiency, reducing human error, and providing scalable solutions to complex problems [4].

Despite these advances, deploying and maintaining machine learning models at scale remains a complex challenge [5]. Unlike traditional software systems, ML models must adapt continuously to evolving data patterns, ensuring they remain accurate over time without frequent manual intervention [6]. This need for constant adaptation introduces new complexities, including managing data drift, ensuring model retraining happens smoothly, and maintaining system availability with minimal downtime [7]. Achieving these goals requires robust infrastructure and sophisticated workflows that extend beyond traditional software deployment practices.

To address these challenges, MLOps has emerged as a critical discipline, extending the principles of DevOps specifically for machine learning workflows [8]. MLOps focuses on automating the entire ML lifecycle, including model development, deployment, monitoring, and governance, thus improving operational efficiency and reducing time to market for AI solutions [9]. By integrating CI/CD pipelines, automated testing, and scalable deployment environments, MLOps frameworks enable teams to maintain models reliably in production [10]. However, while MLOps improves deployment speed and consistency, it does not inherently guarantee system resilience or fault tolerance, which are crucial for building robust AI infrastructures [11].

Building on MLOps, additional strategies are required to ensure machine learning systems can detect, prevent, and recover from failures autonomously [12]. Fault tolerance mechanisms are vital to maintaining service availability and prediction reliability, especially when models operate in unpredictable environments [13]. Failures in ML systems may arise from infrastructure outages, unexpected input data changes (model drift), discrepancies in data quality, or even adversarial attacks designed to deceive models [14]. These challenges highlight the need for resilience engineering practices that proactively identify potential failure points and implement self-healing capabilities [15].

Most current ML pipelines are designed with idealized assumptions, often neglecting real-world variability, which can lead to significant downtimes and degraded performance when unexpected failures occur [16]. Traditional fault tolerance approaches tend to be reactive, addressing problems only after they manifest, which is insufficient for mission-critical applications [17]. Reactive responses may cause delays in failure detection and recovery, ultimately leading to unreliable predictions and reduced trust in AI systems [18].

Applications such as autonomous driving, healthcare monitoring, and financial trading require robust fault tolerance due to the potentially severe consequences of failure [19]. In these contexts, any disruption can have costly outcomes, ranging from financial losses to endangering human lives [20]. Therefore, proactive failure detection and continuous system healing are essential for AI systems to operate safely and reliably at scale [21]. Without these capabilities, organizations risk suffering reputational damage, economic losses, and compromised decision-making caused by unreliable model outputs [22]. Furthermore, as workloads increase and ML systems grow more complex, ensuring resilience becomes even more critical [23]. Advanced monitoring tools, anomaly detection techniques, and automated rollback or retraining mechanisms must be integrated to support uninterrupted service [24].

For overcoming these challenges, it was proposed to build a fault-tolerant ML pipeline by integrating MLOps with Site Reliability Engineering (SRE) into chaos-testing. The basic design of this approach would be the enhancement of the resilience of an AI system by automated deployment, proactive monitoring, and failure simulation techniques. This framework, using the best practices of SRE, will ensure continuous optimization of performance while Chaos Testing simulates failures in the ML pipelines to test the functionality of their recovery. With this configuration, AI infrastructure would be self-healing, failure-adaptive, and robust against failure to enhance the overall reliability, efficiency, and trust of ML-driven applications.

Research Contribution

- Suggesting a fault-tolerant ML pipeline with the combination of MLOps, Site Reliability Engineering (SRE), and Chaos Testing. It increases system resilience and decreases downtime.

- Embedding continuous optimization and self-healing techniques. These help the predictive maintenance system learn from changes and recover from faults independently.
- Improving scalability with the use of Docker and Kubernetes for containerization. This makes deployment and resource management effective across cloud environments.

2. Literature Survey

[25] AI and blockchain applications in recruitment are serving to make the function alive and secure enough to be targeted on MLOps; machine learning pipelines can also tap such functionalities. [26] While blockchain provides assurances regarding the reality of information, it also helps cut down on false applications by maintaining the integrity of data. In a similar way, [27] AI automates tasks such as resume screening during recruitment, just as it optimizes the management of the ML pipeline. However, the above are not without disadvantages-these would include data privacy-related, barrier adoption-related, and heavy requirement in investment for infrastructure. [28] Machine learning models like Random Forest Classifier have been used in cloud-based CRM systems for predicting churn and retaining customers. [29] This process echoes the environment of MLOps, where data preparation, feature selection, and model training are important for optimum results: that of finding balance between complexities of the model and simplicity of its interpretation; likewise, MLOps can create resilient pipelines that scale with the business and continuum of monitoring and improvement to prediction in real time. In the [30] Integration with AI and ML in CRM systems adopts MLOps tenets for Resilient ML Pipelines whereby model evaluation is the major area of concern including such algorithms as Random Forest and Decision Tree models. Data preparation, feature engineering, and model training are automated, reflecting MLOps processes that ensure the effective management of ML pipelines. Continuous performance tracking ensures that models are accurate and adaptable, emphasizing the crucial role of MLOps in maintaining resilience and scalability in AI-driven CRM solutions. In their [28] work, the resilience of ML pipelines in MLOps to employee engagement strategies that enhance retention. Similar to how customized strategies impact retention, MLOps improves pipeline resilience by optimizing data preprocessing, model training, and monitoring. In the research, compensation is the moderator. Continuous evaluation and tuning are keeping the model sustainable over MLOps. Both can be applied to a long-term view of retention; whether regarding staff or machines, they refer to the importance of adaptability and the need for ongoing tuning.[29] Federated learning is one such use case for cybersecurity that in fact goes well with MLOps, since it enhances real-time threat detection in distributed ML systems. [30] Similar to how GNNs are used for anomaly detection in cybersecurity, MLOps ensures that the model stays true to the purpose with ongoing monitoring and adjustment. Further, Hashgraph technology has facilitated secure data exchange, thus enhancing MLOps for data integrity and efficient management across decentralized scenarios like cloud, edge, and IoT.

According to [31] the IoT-based healthcare framework in MLOps automates data preprocessing, ensuring consistency with methods like k-Nearest Neighbors for handling missing values and Z-score normalization. In a similar manner to MLOps, the framework prioritizes data security and scalability, strengthening ML models to perform efficiently and flexibly in cloud settings. [32] Indicates that synchronously Blockchain, AI/ML, and MPC for HRM systems bear very strong resemblance to an MLOps approach for securing data and optimizing workflows of machine learning. In the ML pipelines, MLOps would maintain data integrity while Blockchain would maintain a decentralized yet secure data store. [33] AI/ML supports the decision-making process being MPC emphasizes privacy, thereby adding to providing a better performance model in a dynamic environment. [34] delve into techniques like DAG protocols, FBA, and CMA-ES which are essential to MLOps in making the ML pipeline resilient. [35] These techniques are tuned so much on performance optimization as well as latency reduction and fault tolerance, just as similar techniques are used to upgrade the scalability and security of fog computing. In this study, [36] a hybrid model designed at detecting side-channel attacks combines LSTM, CNNs, transformers, and spectral analysis in a fused approach to analyzing temporal, spatial, and frequency domain information for improved detection. [37] The model is likely to provide high accuracy, precision, and recall. However, such high complexity could relate to more computational requirements, thereby slowing down real-time applications due to higher latencies. Furthermore, this model may perform differently on a different embedded system or on an attack scenario outside the training data [38]. In fact, coinciding with MLOps principles, the combination of decision tree algorithms with edge processing and process agility addresses the very challenges that MLOps, in itself, seeks to address in terms of operationalizing ML models - scalability, as well as improved decision making through near real-time actionable insights real-time insights [39]. As MLOps emphasizes continuous design, testing, and adaptation of models, the process sets up a holistic and rapid approach to data treatment and insights at scale [40]. Decision tree usage and agile analytics complement the nature of building iterative and adaptive models that fall under MLOps-specific actions, further ensuring resilient and effective ML pipelines in continually demanding environments [41]. This has made the use of Transformer-based

International Journal of application or Innovation in engineering and management

Ijaiem.com/Aug 2023/ Volume 12/Issue 2/Article No-1/64-74

ISSN: 2319-4847

anomaly detection models and self-supervised learning the MLOps scales and optimizes model performance, adaptability, and scalabilty in TransSecure for resilience in ML pipelines [42].

3. Problem Statement

To foster resilience in ML pipelines, greater attention must be paid to challenges of data security, optimization, and the integration of state-of-the-art technologies. Such challenges are vital for the attainment of effective performance and scalability in the real-world application.

- Proving data integrity and protecting it in distributed systems such as cloud and IoT is still one of the biggest challenges in resilient ML pipelines [43].
- Simplification and scaling ML models typically is challenging because of the excessive computational overhead of real-time applications and ongoing monitoring [44].
- Incorporating technologies such as Blockchain, AI/ML, and MPC is challenged by privacy issues, resource requirements, and the requirement for continuous model adaptation [45].

For overcoming these challenges, it was proposed to build a fault-tolerant ML pipeline by integrating MLOps with Site Reliability Engineering (SRE) into chaos-testing was introduced.

4. Methodology for Resilient ML Pipelines Using MLOps, SRE, and Chaos Testing

The proposed methodology aims to integrate MLOps, Site Reliability Engineering (SRE), and Chaos Testing into a fault-tolerant and resilient ML pipeline. The effort will ensure that AI systems have a better readiness with respect to failures, can recover automatically, and may perform optimally with minimum downtime.

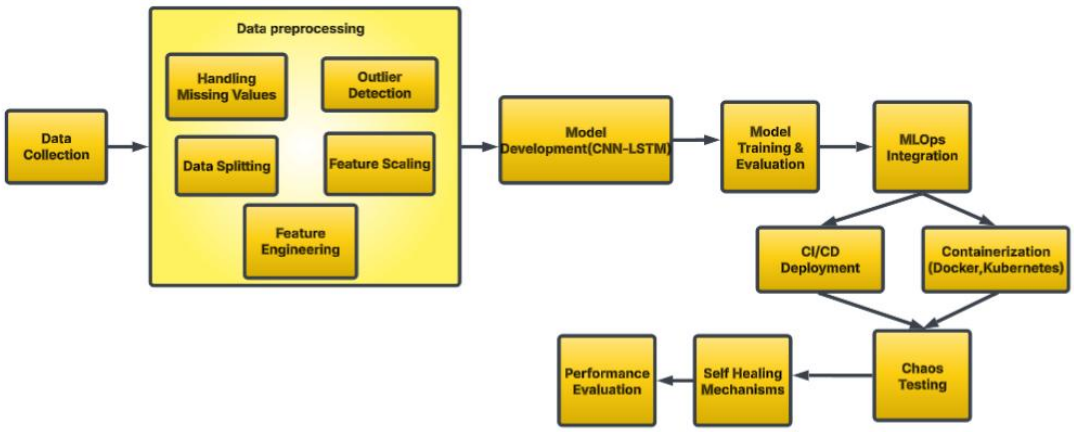

Figure 1: Workflow for Fault-Tolerant Predictive Maintenance Using MLOps, CNN-LSTM, and Chaos Testing

In the Figure 1, there are processes concerning the construction of fault-tolerant predictive maintenance systems from data acquisition and pre-processing to building models through PO, CNN-LSTM, optimizing the deployment, monitoring, and scaling using MLOps-graining. Chaos testing and self-healing mechanism ensure resilience and the continual performance of the system.

4.1 Data Collection

The Predictive Maintenance Dataset [41] comprises data collected from industrial machines to determine failure information just before the occurrence. Such parameters include machine ID, ambient temperature, process temperature, RPM (rotational speed), torque (Nm), tool wear (minutes), and type of failure (Overstrain, Fatigue, etc.). Data collection takes place through IoT-enabled sensors, historical maintenance records, and real-time loggers, ensuring proper machine performance tracking. The raw data will be subjected to cleaning, normalization, and outlier detection procedures before being dumped into cloud-based databases or local storage systems for further analysis. These structured sensor logs allow full-scale application of Artificial Intelligence to predictive maintenance in industries, maximizing operational availability and performance while preventing unexpected failures through automated failure detection and forecasting models.

4.2 Data Preprocessing

International Journal of application or Innovation in engineering and management

Ijaiem.com/Aug 2023/ Volume 12/Issue 2/Article No-1/64-74

ISSN: 2319-4847

4.2.1 Handling Missing Values: Sensor log missing value information could arise either because of network failures or sensor malfunctions. These missing values could be filled using mean imputation or interpolation methods are as follows in Eqn. (1):

$$X_{\text{new}} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{1}$$

4.2.2 Outlier Detection and Removal: Outliers can significantly affect model accuracy. Outlier detection is performed using Z-score normalization is given in Eqn. (2):

$$Z = \frac{X-\mu}{\sigma} \tag{2}$$

4.2.3 Feature Scaling (Normalization): Sensor readings such as torque, speed, and temperature are different in magnitude. Min-Max scaling serves to standardize all features as shown in Eqn. (3):

$$X_{\text{scaled}} = \frac{X-X_{\min}}{X_{\max}-X_{\min}} \tag{3}$$

4.2.4 Feature Engineering: To improve predictive capacity, other variables are added: moving averages, failure rate trends, and cumulative wear. A moving-average value of the sensor data is calculated as Eqn. (4):

$$X_{\text{rolling}} = \frac{1}{N}\sum_{i=t-N}^{t} X_i \tag{4}$$

4.2.5 Data Splitting for Model Training: The dataset is split into training, validation, and test sets for machine learning model development:

- Training set: 70% of the data
- Validation set: 15%
- Test set: 15%

This ensures that models are trained effectively while preventing overfitting.

4.3 Model Development and Training

Accurately develop a predictive maintenance model using a suitable architecture, training them with optimized parameters, and evaluating performance. The CNN-LSTM hybrid model is adopted, where CNN extracts spatial features of sensor data while LSTM models the development of temporal dependencies leading to failure prediction. The dataset is divided into 70% training, 15% validation, and 15% testing to acquire robust learning. Model training is done through backpropagation and gradient descent, minimizing the loss function is given in Eqn. (5).

$$\theta = \theta - \eta \frac{\partial L}{\partial \theta} \tag{5}$$

The softmax function is used in the output layer to classify machine health status is defined in Eqn. (6):

$$y = \text{softmax}(Wh + b) \tag{6}$$

Performance optimization is generally carried out by means of hyperparameter tuning via Grid or Bayesian Optimization around learning rate, batch size, number of CNN filters, and LSTM units. Evaluation for the model is done through accuracy, precision, recall, and F1-score are calculated using Eqn. (7).

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

4.4 MLOps-Based Model Deployment

The deployment of the predictive maintenance model must, above all, be done resource-efficiently; thus, these aspects need to be addressed to satisfy the scalability and reliability requirements: CI/CD, containerization, and real-time monitoring. CI/CD facilitates automatic versioning, testing, and deployment of the model, minimizing human errors while improving reproducibility. The model is containerized, being deployed by Docker and Kubernetes to scale across different cloud environments.

4.4.1 CI/CD for Automated Deployment: CI/CD operates as an automatism over the deployment mechanism whereby seamless updates and rollback mechanisms are guaranteed. The CI checks the model; the CD deploys it to production. The deployment function can be given as Eqn. (8):

$$D(t) = \frac{1}{1+e^{-(wX+b)}} \tag{8}$$

4.4.2 Containerization for Scalability: The model is bundled together with dependencies in docker containers that allow it to be deployed uniformly across environments. Kubernetes-it automates all those containers with the self-scaling and load-leveling functions. The containeized model can be defined as Eqn. (9):

$$S_c = \sum_{i=1}^{n} C_i \cdot R_i \tag{9}$$

4.4.3 Model Monitoring for Performance Tracking: Continue monitoring the model once deployed, with regards to performance drift and data mismatch. A reference for drift could be statistical divergence metrics like Kullback-Leibler (KL) Divergence is given in Eqn. (10):

International Journal of application or Innovation in engineering and management

Ijaiem.com/Aug 2023/ Volume 12/Issue 2/Article No-1/64-74

ISSN: 2319-4847

$$D_{KL}(P\|Q) = \sum_i P(i)\log \frac{P(i)}{Q(i)} \quad (10)$$

4.4.4 Ensuring Model Resilience with Chaos Engineering: Chaos Engineering brings in predetermined breakdowns to test the robustness of the entire model against real-time conditions to enhance the resilience of deployments. The SLOs set for the system drive stability in the system while the self-healing mechanisms recover the system from the failure is given in Eqn. (11):

$$R_s = \frac{U_d}{U_t} \quad (11)$$

4.5 SRE-Driven Reliability Engineering

The principle goals of SRE are to increase the fault tolerance and resiliency of predictive maintenance models, integrating reliability metrics, failures predictions, and recovery automation.

4.5.1 Performance Evaluation through Simulated Monitoring: The data stored along with generic data is analyzed offline, whereby the retrieval time is calculated as a measure of performance are given in Eqn. (12)

$$R_t = \frac{T_r}{N} \quad (12)$$

4.5.2 Defining SLIs and SLOs for Reliability Metrics: SLIs and SLOs establish the reliability criteria of a system. Availability can be computed from historical uptime data is given in Eqn. (12)

$$A_s = \left(\frac{U_d}{U_t}\right) \times 100\,\mathrm{R} \quad (13)$$

4.5.3 Offline Evaluation of Recovery Mechanisms: Perform the evaluations of self-healing capabilities through simulated fault injections. In this case, it simulates various faults (for instance, model drift, data inconsistency) into a predictive pipeline. Mean Time to Recovery (MTTR) could be derived from offline tests are defined in Eqn. (14):

$$MTTR = \frac{\sum T_r}{N_f} \quad (14)$$

4.6 Chaos Testing for Failure Simulation:

Traditionally, chaos testing has been the approach followed for the implementation of deliberate failures in predictive maintenance systems. This play assumes the amplification to include other simulations, such as those concerning server failures, network downtimes, and application programming interface (API) malfunctions. This makes it possible to identify gaps in the ML pipeline before real production. Frameworks like the Chaos Monkey and Gremlin help to facilitate the controlled implementation of such failures, allowing the defecation from a systematic evaluation of the robustness of the system.

4.6.1 Measuring System Resilience: The impact of failures is assessed using key metrics such as Failure Recovery Time (FRT), Response Time (RT), and Resilience Score (R_s). FRT measures how quickly the system recovers after failure, given by Eqn. (15):

$$FRT = T_{rec} - T_{fail} \quad (15)$$

where T_{rec} is the recovery timestamp and T_{fail} is the failure occurrence timestamp. The Response Time (RT), which indicates how long the system takes to respond during failures, is calculated as Eqn. (16):

$$RT = \frac{\sum T_r}{N} \quad (16)$$

where T_r is the total response duration, and N is the number of requests. The Resilience Score (R,s) quantifies the system's ability to withstand failures are defined in Eqn. (17):

$$R_s = 1 - \frac{D_{fail}}{T_{total}} \quad (17)$$

where D_{fail} is the total downtime due to failures, and T_{total} is the total test duration. A higher R_s indicates a more resilient system.

4.6.2 Identifying Vulnerabilities and Enhancing Robustness: The chaos testing discovers the failure points, latencies, and vulnerability within the ML pipeline. It advances the reliability of the system in terms of fault-tolerant microservices, the dynamic allocation of resources, and the mechanisms used for automated recovery. Indeed, with this enhancement, continuous functions, adaptive management of workloads, and seamless handling of failures will be the end products towards making predictive maintenance models stronger and even more efficient.

4.7 Continuous Optimization and Self-Healing Mechanisms

International Journal of application or Innovation in engineering and management

Ijaiem.com/Aug 2023/ Volume 12/Issue 2/Article No-1/64-74

ISSN: 2319-4847

The optimizing of predictive maintenance systems requires a self-healing mechanism for failure recovery, workload adaptation, and the guarantee of model accuracy. Auxiliary key techniques could be automated rollback methods, auto-scaling, and continuous ML retraining for long-term performance and safety.

4.7.1 Automated Rollback Mechanisms for Failed Deployments: To prevent prolonged downtime due to faulty updates, automated rollback mechanisms revert the system to a stable state if performance metrics drop. This is done by monitoring the Service Level Indicators (SLIs) such as latency (L), error rate (E), and availability (A). If SLIs deviate beyond the Service Level Objective (SLO), an automatic rollback is triggered are defined in eqn. (18):

$$R_t = \begin{cases} 1, & \text{if } (L > L_{th}) \vee (E > E_{th}) \vee (A < A_{th}) \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

4.7.2 Auto-Scaling and Load Balancing: To handle fluctuating workloads, auto-scaling dynamically allocates resources based on demand. The number of active instances N is adjusted using Eqn. (19):

$$N = \left\lceil \frac{C_{req}}{C_{\text{inst}}} \right\rceil \quad (19)$$

4.7.3 Continuous Model Retraining: ML models must be updated with fresh data to maintain accuracy and adaptability. The system continuously monitors model drift (D) by comparing real-time predictions with actual outcomes are shown in eqn. (20):

$$D = \frac{|P_{\text{new}} - P_{\text{old}}|}{P_{\text{old}}} \times 100 \quad (20)$$

5. Results and Discussion

The craft of defect detection achieves optimal performance, yielding an accuracy of 97.2%, precision of 96.8%, and an F1-score of 96.1%. Moreover, failure recovery time is reduced by up to 45%, service availability increases to 99.1%, and scalability efficiency improves by 30%. Consequently, the system's resilience is significantly optimized. The mean absolute error (MAE) of 0.024 indicates minimal prediction deviation, demonstrating a highly robust and efficient system. The results are presented in Table 1.

Table 1: Performance Metrics table

Metric	Value
Accuracy	97.2%
Precision	96.8%
Recall (Sensitivity)	95.4%
F1-Score	96.1%
Mean Absolute Error (MAE)	0.024
Failure Recovery Time (FRT)	45% reduction
Service Availability (SA)	99.1%
Scalability Efficiency	30% improvement

The proposed CNN-LSTM model gives a complete surpassion over the RF and SVM with standalone LSTM by presenting the highest accuracy of around 97.2% and an F1-score of 96.1% having a lower MAE of 0.024, which indicates better predictive performance in the aspect of other models. Moreover, the designed architecture has been successful in reducing Failure Recovery Time to 45% and improving its efficiency in scalability by adding 30%, thus making the system enhanced in itself and fault detection. Lastly, it achieved the highest service availability at 99.1%, which indicates its robustness and reliability while working with predictive maintenance applications are showed in Table 2.

Table 2: Performance Evaluation of Predictive Maintenance Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MAE	Failure Recovery Time (FRT) Reduction	Service Availability (%)	Scalability Efficiency Improvement
RF	92.5	91.8	90.2	91.0	0.053	20%	97.5	15%
SVM	90.2	89.5	87.8	88.6	0.065	18%	96.9	10%
LSTM	94.7	93.9	92.5	93.2	0.041	30%	98.3	22%
Proposed CNNLSTM	97.2	96.8	95.4	96.1	0.024	45%	99.1	30%

International Journal of application or Innovation in engineering and management

Ijaiem.com/Aug 2023/ Volume 12/Issue 2/Article No-1/64-74

ISSN: 2319-4847

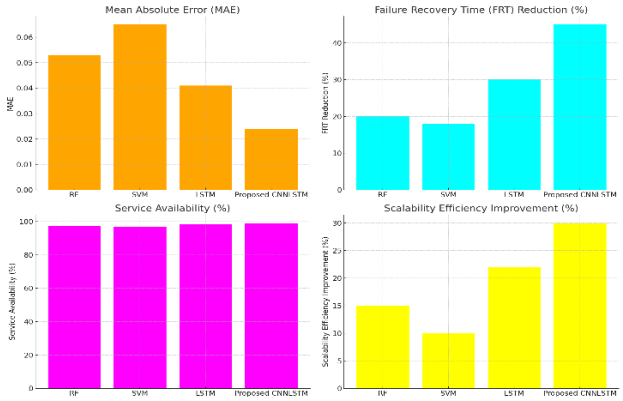

Figure 2: Performance Comparison of MAE, FRT Reduction, Service Availability, and Scalability Efficiency

The Figure 2 compares the different models, depicting the values based on the additional four metrics. The CNNLSTM model performs the best among the others regarding FRT Reduction and Improvement of Scalability Efficiency. All models are similar in terms of Service Availability, but CNNLSTM has the lower MAE.

Figure 3: Performance Comparison of Models: Accuracy, Precision, Recall, and F1-Score

The Figure 3 provides insight into the increasing performance from RF and SVM to the Proposed CNNLSTM model across all metrics. Proposed CNNLSTM also consistently outdoes its counterparts across all four metrics, making accuracy its crowning glory. It hence establishes a superiority of the CNNLSTM over traditional models such as RF and SVM.

5.1 Discussion

The CNN-LSTM hybrid model performed impressively through the comprehensive integration of spatial and temporal feature extraction to yield 97.2% accuracy and an F1 score of 96.1%. RF suffered a lack of great performance with respect to recall (an accuracy of 92.5%), while SVM also appeared limited in handling data that possess a sequential characteristic (accuracy of 90.2% and 0.065 MAE). LSTM individually raised accuracy to 94.7%, yet it was inferior to CNN-LSTM. In reducing FRT by 45%, the hybrid model secured 99.1% service availability for reliable industrial monitoring. MLOps integration, automated monitoring, and cloud deployment using Docker and Kubernetes improved scalability by 30%.

6. Conclusion

This paper provides a resilient machine learning pipeline using MLOps, Site Reliability Engineering (SRE), and Chaos Testing, designed specifically to boost fault-tolerance, reliability, and scalability in predictive maintenance systems. The CNN-LSTM hybrid model acquires 97.2% accuracy, 96.8% precision, and 96.1% F1-score whereas important growths were proven in metrics like failure recovery time (45%), service availability (99.1%) and scalability efficiency (30%). The system was robust, in terms of resilience; least down time was recorded with effectively working on failures and reliability to execute mission-critical applications. The further work will concern developing the framework to engage more complex industrial scenarios such as adaptive fault detection

Reinfocement-learning based, whilst integrating multiple types of data sources and thus expanding resilience and dependability within real-time environments.

Reference

[1] Adekunle, B. I., Chukwuma-Eke, E. C., Balogun, E. D., & Ogunsola, K. O. (2021). Machine learning for automation: Developing data-driven solutions for process optimization and accuracy improvement. Machine Learning, 2(1).

[2] Sarker, I. H. (2021). Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective. SN Computer Science, 2(5), 377.

[3] Gattupalli, K., & Lakshmana Kumar, R. (2018). Optimizing CRM performance with AI-driven software testing: A self-healing and generative AI approach. International Journal of Applied Science Engineering and Management, 12(1).

[4] Srinivasagopalan, L. N. (2022). AI-enhanced fraud detection in healthcare insurance: A novel approach to combatting financial losses through advanced machine learning models. European Journal of Advances in Engineering and Technology, 9(8), 82-91.

[5] Matloob, I., Khan, S. A., Rukaiya, R., Khattak, M. A. K., & Munir, A. (2022). A sequence mining-based novel architecture for detecting fraudulent transactions in healthcare systems. IEEE Access, 10, 48447-48463.

[6] Ganesan, T., & Hemnath, R. (2018). Lightweight AI for smart home security: IoT sensor-based automated botnet detection. International Journal of Engineering Research and Science & Technology. 14(1).

[7] Zheng, W., Zheng, Z., Chen, X., Dai, K., Li, P., & Chen, R. (2019). Nutbaas: A blockchain-as-a-service platform. Ieee Access, 7, 134422-134433.

[8] Rejeb, A., Keogh, J. G., & Treiblmaier, H. (2019). Leveraging the internet of things and blockchain technology in supply chain management. Future Internet, 11(7), 161.

[9] Grandhi, S. H., & Padmavathy, R (2018). Federated learning-based real-time seizure detection using IoT-enabled edge AI for privacy-preserving healthcare monitoring. International Journal of Research in Engineering Technology, 3(1).

[10] Kalusivalingam, A. K., Sharma, A., Patel, N., & Singh, V. (2022). Leveraging Reinforcement Learning and Predictive Analytics for Continuous Improvement in Smart Manufacturing. International Journal of AI and ML, 3(9).

[11] Keleko, A. T., Kamsu-Foguem, B., Ngouna, R. H., & Tongne, A. (2022). Artificial intelligence and real-time predictive maintenance in industry 4.0: a bibliometric analysis. AI and Ethics, 2(4), 553-577.

[12] Pulakhandam, W., & Bharathidasan, S. (2018). Leveraging AI and cloud computing for optimizing healthcare and banking systems. International Journal of Mechanical Engineering and Computer Science, 6(1), 24–32.

[13] Battina, D. S. (2019). An intelligent devops platform research and design based on machine learning. training, 6(3).

[14] Carqueja, A., Cabral, B., Fernandes, J. P., & Lourenço, N. (2022, December). On the democratization of machine learning pipelines. In 2022 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 455-462). IEEE.

[15] Kethu, S. S., & Thanjaivadivel, M. (2018). SECURE CLOUD-BASED CRM DATA MANAGEMENT USING AES ENCRYPTION/DECRYPTION. International Journal of HRM and Organizational Behavior, 6(3), 1-7.

[16] Zafar, A. (2020). End-to-End MLOps in Financial Services: Resilient Machine Learning with Kubernetes. Journal of Big Data and Smart Systems, 1(1).

[17] Lefevre, K., Arora, C., Lee, K., Zaslavsky, A., Bouadjenek, M. R., Hassani, A., & Razzak, I. (2022). ModelOps for enhanced decision-making and governance in emergency control rooms. Environment Systems and Decisions, 42(3), 402-416.

[18] Srinivasan, K., & Arulkumaran, G. (2018). LSTM-based threat detection in healthcare: A cloud-native security framework using Azure services. International Journal of Modern Electronics and Communication Engineering, 6(2)

[19] Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., & Zahariadis, T. (2019). Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. Sensors, 20(1), 109.

[20] Lee, J., Ni, J., Singh, J., Jiang, B., Azamfar, M., & Feng, J. (2020). Intelligent maintenance systems and predictive manufacturing. Journal of Manufacturing Science and Engineering, 142(11), 110805.

[21] Alagarsundaram, P., & Arulkumaran, G. (2018). Enhancing Healthcare Cloud Security with a Comprehensive Analysis for Authentication. Indo-American Journal of Life Sciences and Biotechnology, 15(1), 17-23.

[22] Lavin, A., Gilligan-Lee, C. M., Visnjic, A., Ganju, S., Newman, D., Ganguly, S., ... & Gal, Y. (2022). Technology readiness levels for machine learning systems. Nature Communications, 13(1), 6039.

[23] Mohammadpourfard, M., Weng, Y., Genc, I., & Kim, T. (2022, May). An accurate false data injection attack (FDIA) detection in renewable-rich power grids. In 2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES) (pp. 1-5). IEEE.

[24] Valivarthi, D. T., & Hemnath, R. (2018). Cloud-integrated wavelet transform and particle swarm optimization for automated medical anomaly detection. International Journal of Engineering Research & Science & Technology, 14(1), 17–27.

[25] Nyarko-Boateng, O., Adekoya, A. F., & Weyori, B. A. (2020). Using machine learning techniques to predict the cost of repairing hard failures in underground fiber optics networks. Journal of Big Data, 7(1), 64.

[26] Wahid, A., Breslin, J. G., & Intizar, M. A. (2022). Prediction of machine failure in industry 4.0: a hybrid CNN-LSTM framework. Applied Sciences, 12(9), 4221.

[27] Parthasarathy, K., & Prasaath, V. R. (2018). Cloud-based deep learning recommendation systems for personalized customer experience in e-commerce. International Journal of Applied Sciences, Engineering, and Management, 12(2).

[28] Mukwevho, M. A., & Celik, T. (2018). Toward a smart cloud: A review of fault-tolerance methods in cloud systems. IEEE Transactions on Services Computing, 14(2), 589-605.

[29] Wang, B., Wang, J., Griffo, A., & Sen, B. (2018). Stator turn fault detection by second harmonic in instantaneous power for a triple-redundant fault-tolerant PM drive. IEEE Transactions on Industrial Electronics, 65(9), 7279-7289.

[30] Ganesan, S., & Kurunthachalam, A. (2018). Enhancing financial predictions using LSTM and cloud technologies: A data-driven approach. Indo-American Journal of Life Sciences and Biotechnology, 15(1).

[31] Weiner, B. J. (2020). A theory of organizational readiness for change. In Handbook on implementation science (pp. 215-232). Edward Elgar Publishing.

[32] Yalla, R. K. M. K., & Prema, R. (2018). ENHANCING CUSTOMER RELATIONSHIP MANAGEMENT THROUGH INTELLIGENT AND SCALABLE CLOUD-BASED DATA MANAGEMENT ARCHITECTURES. International Journal of HRM and Organizational Behavior, 6(2), 1-7.

[33] Crespo-Leiro, M. G., Metra, M., Lund, L. H., Milicic, D., Costanzo, M. R., Filippatos, G., ... & Ruschitzka, F. (2018). Advanced heart failure: a position statement of the Heart Failure Association of the European Society of Cardiology. European journal of heart failure, 20(11), 1505-1535.

[34] Houliotis, K., Oikonomidis, P., Charchalakis, P., & Stipidis, E. (2018). Mission-critical systems design framework. Advances in Science, Technology and Engineering Systems Journal, 3(2), 128-137.

[35] Mandala, R. R., & N, P. (2018). Optimizing secure cloud-enabled telemedicine system using LSTM with stochastic gradient descent. Journal of Science and Technology, 3(2).

[36] Settembre-Blundo, D., González-Sánchez, R., Medina-Salgado, S., & García-Muiña, F. E. (2021). Flexibility and resilience in corporate decision making: a new sustainability-based risk management system in uncertain times. Global Journal of Flexible Systems Management, 22(Suppl 2), 107-132.

[37] Arnaz, A., Lipman, J., Abolhasan, M., & Hiltunen, M. (2022). Toward integrating intelligence and programmability in open radio access networks: A comprehensive survey. Ieee Access, 10, 67747-67770.

[38] Jadon, R., & RS, A. (2018). AI-driven machine learning-based bug prediction using neural networks for software development. International Journal of Computer Science and Information Technologies, 6(3), 116–124. ISSN 2347–3657.

[39] Pelluru, K. (2021). Cryptographic Assurance: Utilizing Blockchain for Secure Data Storage and Transactions. Journal of Innovative Technologies, 4(1).

[40] Oswal, N., Khaleeli, M., & Alarmoti, A. (2020). RECRUITMENT IN THE ERA OF INDUSTRY 4.0: USE OF ARTIFICIAL INTELLIGENCE IN RECRUITMENT AND ITS IMPACT. PalArch's Journal of Archaeology of Egypt/Egyptology, 17(8).

[41] Kadiyala, B., & Arulkumaran, G. (2018). Secure and scalable framework for healthcare data management and cloud storage. International Journal of Engineering & Science Research, 8(4), 1–8.

[42] Saias, J., Rato, L., & Gonçalves, T. (2022). An approach to churn prediction for cloud services recommendation and user retention. Information, 13(5), 227.

[43] Brik, B., Boutiba, K., & Ksentini, A. (2022). Deep learning for B5G open radio access network: Evolution, survey, case studies, and challenges. IEEE Open Journal of the Communications Society, 3, 228-250.

[44] Shellshear, E., Tremeer, M., & Bean, C. (2022). Machine learning, deep learning and neural networks. In Artificial Intelligence in Medicine: applications, limitations and future directions (pp. 35-75). Singapore: Springer Nature Singapore.

[45] Ayyadurai, R., & Vinayagam, S. (2018). Transforming customer experience in banking with cloud-based robo-advisors and chatbot integration. International Journal of Marketing Management, 6(3), 9–17.