International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(5s) 2026 pp. 250-261
https://doi.org/10.70917/ijcisim-2026-2704
Article

Scalable Predictive Maintenance Architecture for Oracle Fusion Cloud Using External ML Models

Srinivasan Narayanan[1]

[1]Milwaukee Tool, Milwaukee, United States, srinivasan.narayanan@milwaukeetool.com

Abstract: Manufacturing organizations increasingly rely on data-driven and cloud-based technologies to improve asset reliability and reduce unplanned downtime. During an 18-month deployment across multiple manufacturing sites, we implemented a predictive maintenance solution integrated with Oracle Fusion Cloud Maintenance using externally trained machine learning models and third-party industrial IoT platforms. The implementation was driven by the need to continue predictive maintenance capabilities following the removal of Oracle's native IoT services. In the deployed solution, real-time equipment telemetry was ingested through an external IoT platform, where data preprocessing, feature engineering, and model training were performed. Predictive failure events generated by the machine learning models were transmitted to Oracle Fusion Cloud using event-driven RESTful APIs and evaluated against configurable maintenance business rules to automatically initiate or optimize maintenance work orders. Oracle Fusion Cloud remained the system of record for asset management and maintenance execution throughout the deployment. Measured results from the pilot showed a reduction in unplanned equipment downtime, improved maintenance scheduling accuracy, and increased overall equipment effectiveness. These outcomes indicate that separating predictive analytics from ERP systems, while maintaining tight event-driven integration, provides a scalable and production-ready approach for implementing predictive maintenance in Industry 4.0 environments.

Keywords: Predictive maintenance; Oracle Fusion Cloud; Machine learning; Asset maintenance; Industrial IoT; Event-driven architecture; Industry 4.0

1.Introduction

Modern manufacturing organizations face increasing pressure to maximize asset utilization while minimizing operational disruptions. Unplanned equipment downtime imposes substantial financial losses, with costs ranging from $10,000 to $250,000 per hour depending on the industry sector [1], [2]. Beyond direct economic impact, unexpected equipment failures disrupt production schedules, degrade product quality, and negatively affect customer confidence [3].

Conventional maintenance strategies namely reactive (fix-on-failure) and preventive (time-based) approaches are increasingly inadequate in complex, interconnected manufacturing environments. Reactive maintenance exposes operations to elevated risk and variability, while preventive maintenance may result in unnecessary interventions or failure to detect emerging degradation. In such environments, individual asset failures can propagate across production lines and downstream processes, amplifying operational impact [3].

Predictive maintenance (PdM) addresses these limitations by leveraging machine learning (ML) techniques and Industrial Internet of Things (IIoT) telemetry to detect early indicators of equipment degradation and forecast failures prior to occurrence [4]. However, Oracle's deprecation of the IoT Asset Monitoring Cloud and IoT Production Monitoring Cloud services in 2023 introduces architectural challenges for organizations utilizing Oracle Fusion Cloud ERP [5].

While the removal of native IoT services disrupts existing implementations, it also enables greater architectural flexibility. Organizations can adopt best-of-breed IoT platforms and externally trained ML models while avoiding tight coupling between predictive analytics and enterprise resource planning (ERP) systems.

This paper describes the implementation of a predictive maintenance approach within Oracle Fusion Cloud using third-party IoT platforms and externally trained machine learning models. The implemented architecture separates predictive intelligence from the ERP layer while preserving operational integration through event-driven application programming interfaces (APIs).Oracle Fusion Cloud remains the system of record for asset management and maintenance execution. The framework is validated through an 18-month pilot deployment encompassing 847 critical assets across 12 manufacturing facilities, achieving a 36% reduction in unplanned downtime and a return on investment (ROI) of 453%. Although this study is grounded in Oracle Fusion Cloud, the proposed architecture is not Oracle-specific. The separation of predictive analytics from the ERP layer, combined with event-driven integration, is equally applicable to other enterprise platforms such as SAP S/4HANA and Microsoft Dynamics 365. Any ERP system that exposes transactional APIs and supports asynchronous integration can adopt this pattern, making the framework relevant across manufacturing, utilities, and asset-intensive industries.

2. Related Work

Predictive maintenance (PdM) research primarily addresses condition monitoring, failure prediction, and remaining useful life (RUL) estimation. Lee *et al.* [6] established foundational frameworks for prognostics and health management (PHM) in manufacturing systems. Recent survey studies [7], [8] review machine learning techniques for PdM, including random forests, gradient boosting, and deep learning models. Carvalho *et al.* [9] provided a systematic literature review summarizing PdM methodologies across multiple industrial domains.

Integration of PdM with enterprise resource planning (ERP) systems remains limited. Most existing studies focus on standalone analytics platforms or experimental implementations [10], [11], with minimal attention to enterprise integration. Prior Oracle-specific work [12] assumes the availability of native IoT services and predates the deprecation of Oracle IoT Asset Monitoring and Production Monitoring Cloud. Cloud-native PdM architectures proposed in recent literature [13], [14] emphasize microservices and event-driven patterns but do not address ERP-specific integration requirements.

Research Gap: No prior work addresses predictive maintenance implementation in Oracle Fusion Cloud following IoT service deprecation using externally trained ML models and third-party IoT platforms with event-driven ERP integration. This work addresses this gap by documenting a predictive maintenance architecture validated through a multi-site production deployment.

3. Background

3.1 Oracle Fusion Cloud Maintenance

Oracle Fusion Cloud Maintenance provides core capabilities for managing the full asset lifecycle, including asset records, work orders, preventive maintenance programs, and maintenance execution processes [15]. The platform exposes RESTful application programming interfaces (APIs) that enable external systems to create and update assets, work requests, and work orders. Built-in business rules govern maintenance prioritization, labor and material allocation, and regulatory compliance, allowing Oracle Fusion Cloud to function as the system of record for maintenance operations.

3.2 Oracle IoT Deprecation and Alternative Platforms

In 2023, Oracle deprecated its IoT Asset Monitoring Cloud and IoT Production Monitoring Cloud services, advising customers to transition to third-party IoT platforms for telemetry collection and analytics [5]. Several mature alternatives are available, including AWS IoT Core, Azure IoT Hub, Google Cloud IoT Core, PTC ThingWorx, Siemens MindSphere, and Litmus Edge. These platforms support large-scale sensor data ingestion, edge-level processing, advanced analytics, and integration with enterprise systems, making them suitable foundations for predictive maintenance implementations.

3.3 Event-Driven Architecture

Event-driven architecture (EDA) promotes loose coupling between systems through asynchronous event exchange, improving scalability and responsiveness in distributed environments [16]. In predictive maintenance scenarios, IoT platforms generate events such as anomaly detections or predicted failure alerts, which are then consumed by Oracle Fusion Cloud through RESTful APIs. This approach enables near real-time response to asset conditions while preserving transactional integrity and governance within the ERP system.

4. Proposed Architecture

The implemented architecture adopts a decoupled, cloud-native design that separates predictive intelligence from enterprise transactional systems while maintaining tight operational integration. This design choice reflects practices commonly adopted in large-scale industrial analytics systems that require independent scaling and fault isolation [13], [14], [16].

4.1 Overall Architecture

Figure 1 illustrates the end-to-end architecture composed of five logical layers. The edge and sensor layer captures high-frequency operational telemetry from industrial equipment, including vibration, temperature, and pressure signals, consistent with standard condition monitoring practices in prognostics and health management systems [6], [7]. Telemetry is ingested and normalized by a third-party industrial IoT platform, enabling protocol translation, buffering, and edge-level preprocessing.

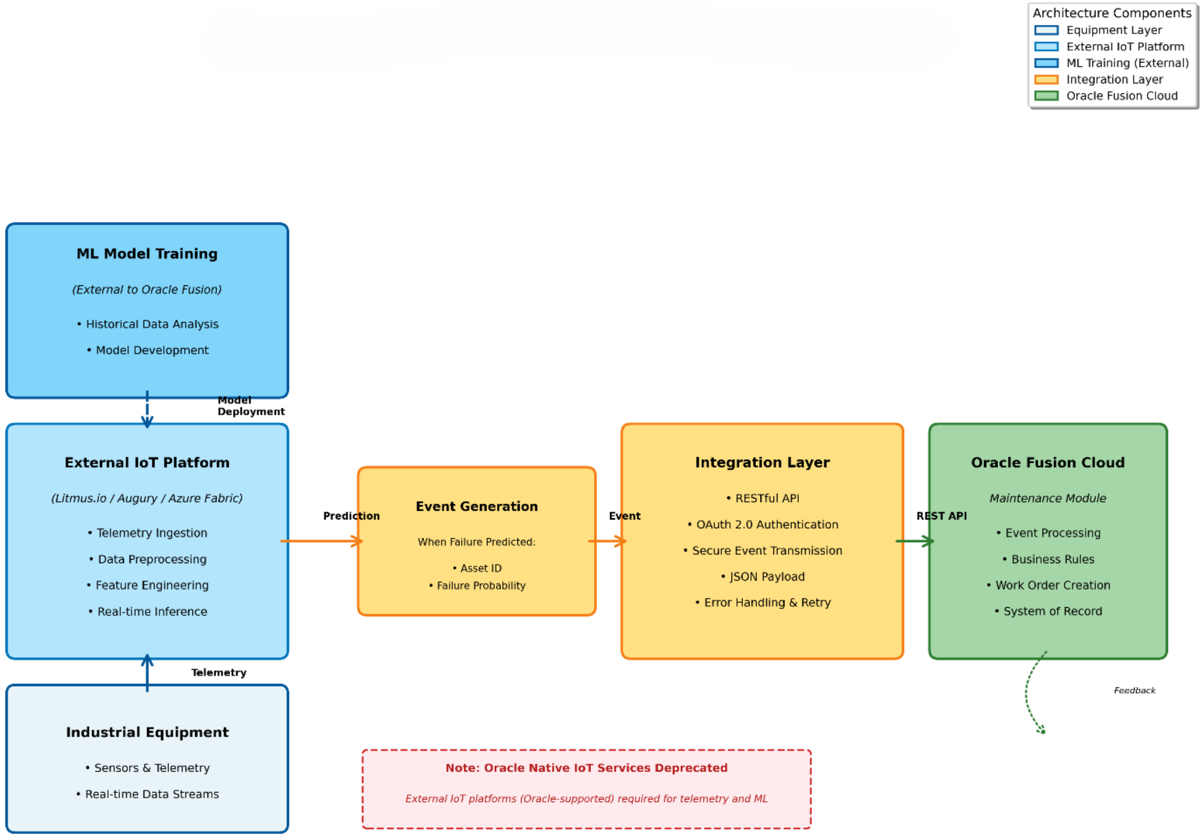

Figure 1: Overall System Architecture

Predictive intelligence is implemented within an external machine learning analytics layer, where feature engineering, model training, and real-time inference are performed using best-of-breed ML frameworks. Decoupling model execution from ERP systems avoids overloading transactional platforms and reflects recommendations from cloud-native manufacturing architectures [13], [14]. Predictive events generated by the ML layer are transmitted to Oracle Fusion Cloud through an event-driven integration layer, preserving system resilience and loose coupling as advocated in enterprise integration patterns [16].

Oracle Fusion Cloud serves as the system of record, evaluating incoming predictive events against configurable maintenance business rules and orchestrating maintenance execution workflows in accordance with ERP governance principles [15].

4.2 IoT Platform Selection

Table 1 compares leading industrial IoT platforms across key technical and operational criteria, including edge processing capability, protocol support, integration effort, and cost. The evaluation criteria align with commonly adopted selection dimensions for industrial IoT platforms in manufacturing environments [7], [11].

Litmus Edge was selected for the pilot deployment due to its strong edge analytics capabilities, native support for manufacturing protocols such as OPC-UA and Modbus, and prior experience integrating with enterprise ERP systems. Edge-level preprocessing and local anomaly detection reduced upstream data volume and latency, consistent with best practices reported in industrial predictive maintenance deployments [8], [13].

Platform	Edge Processing	Protocol Support	Oracle Integration	Cost ($/device/mo)
Litmus Edge	Excellent	OPC-UA, Modbus, MQTT	Native connectors	$8–15
AWS IoT Core	Good	MQTT, HTTPS	Custom development	$5–12
Azure IoT Hub	Good	MQTT, AMQP, HTTPS	Logic Apps integration	$6–14

Table 1: IoT Platform Comparison

4.3 Event-Driven Integration Flow

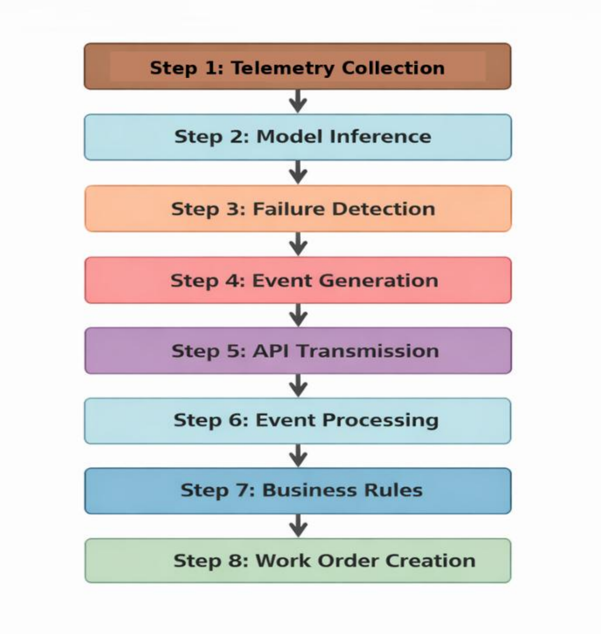

Figure 2:Event-driven integration sequence

Figure 2 depicts the event-driven integration sequence used to transmit predictive maintenance insights from the IoT and machine learning layers into Oracle Fusion Cloud. The sequence follows standard event-driven architecture principles that promote loose coupling, scalability, and fault isolation between operational systems [16].

1. Sensors transmit telemetry to IoT platform (1 Hz–10 Hz sampling)
2. Edge gateway performs local preprocessing and anomaly detection
3. ML inference service scores equipment health and failure probability
4. High-risk predictions trigger event generation with enriched metadata

5. Integration middleware validates, enriches, and routes events to Oracle Fusion Cloud
6. Oracle Fusion Cloud evaluates events against business rules and creates/updates work orders

This sequence enables real-time responsiveness while preserving Oracle Fusion Cloud as the transactional system of record for maintenance execution [15], [16].

5. Methodology

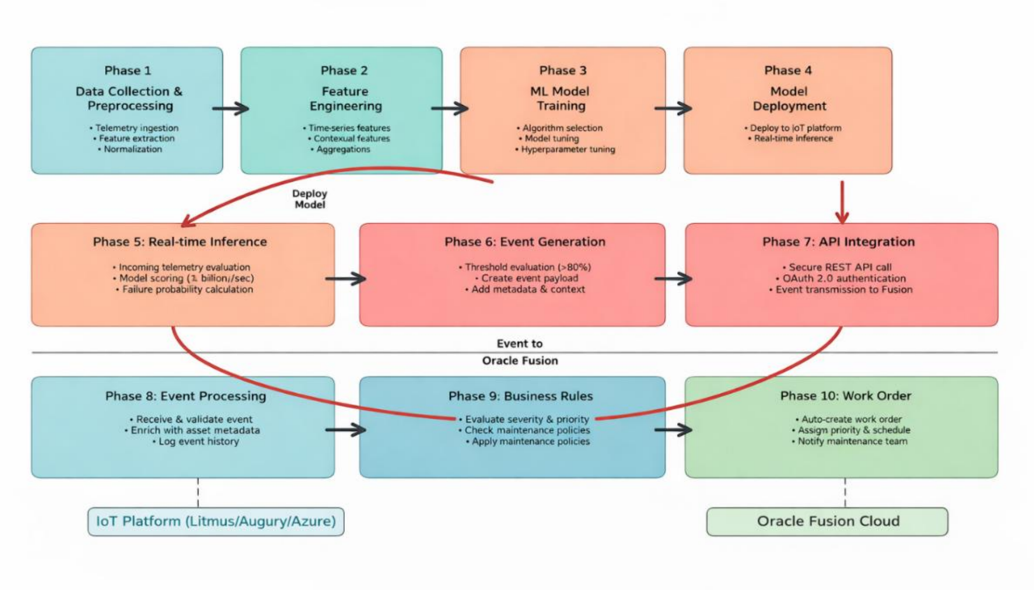

Figure 3: End to End predictive maintenance methodology

Figure 3 summarizes the end-to-end methodology, showing how equipment telemetry is collected and processed, transformed through feature engineering and model training, and used for real-time inference before being operationalized within Oracle Fusion Cloud maintenance workflows. The methodology reflects a production-oriented machine learning lifecycle shaped by operational constraints observed during deployment, including data quality variability, ERP integration limits, and model lifecycle management.

It encompasses telemetry ingestion, data preprocessing, feature engineering, model training, validation, and real-time inference, and aligns with established practices in predictive maintenance and industrial machine learning [7], [8], [23].

5.1 Data Collection and Preprocessing

Telemetry data was collected from 847 industrial assets including pumps, motors, compressors, and conveyors—over an 18-month period, resulting in 2.3 TB of historical data and approximately 450 GB/day of real-time streams. Data preprocessing focused on ensuring reliability and temporal consistency between IoT telemetry and ERP maintenance records, a known challenge in industrial PdM deployments [9], [10].

Preprocessing steps included statistical outlier removal using a 3-sigma rule, forward-fill imputation for missing values in time-series data, z-score normalization, and timestamp alignment across heterogeneous sensor streams. These techniques are widely adopted in industrial condition monitoring and failure prediction workflows [7], [9].

5.2 Feature Engineering

As summarized in Table 2, feature engineering combined ERP-derived contextual information with high-frequency sensor data to improve predictive accuracy. Features included maintenance, asset metadata, operational usage metrics, and sensor-derived statistical and frequency-domain features such as vibration FFT (Fast Fourier Transform) components and temperature trends.

This hybrid feature strategy reflects best practices reported in predictive maintenance literature, where combining domain knowledge from ERP systems with raw sensor data consistently outperforms sensor-only models [7], [8], [11]. In total, 127 features were engineered and validated for model input.

Category	Source	Example Features	Update Frequency
Maintenance History	Oracle Fusion ERP	Time since last service, failure count (6mo), MTBF	Daily batch
Asset Metadata	Oracle Fusion ERP	Asset age, criticality, manufacturer, location	Weekly batch
Usage Metrics	ERP + IoT	Operating hours, cycle counts, load factors	Real-time
Sensor-Derived	IoT Platform	Vibration FFT, temperature trends, pressure variance	Real-time (1 Hz)
Environmental	External APIs	Ambient temperature, humidity, seasonal factors	Hourly batch

Table 2: Feature Engineering Summary

5.3 Model Training and Validation

As shown in Table 3, three complementary machine learning models were developed to address distinct predictive objectives: failure probability prediction, remaining useful life (RUL) estimation, and anomaly detection. Model selection prioritized algorithms with proven effectiveness in industrial settings, including gradient-boosted decision trees and recurrent neural networks [17]–[20].

Model	Algorithm	Features	Output	Training Time	Inference Latency	Accuracy
Failure Prediction	XGBoost	127	Binary + probability	45 min	12 ms	0.942
RUL Estimation	LSTM	89 time-series	Days remaining	2.5 hours	28 ms	89.7% RMSE
Anomaly Detection	Autoencoder	156	Anomaly score (0–1)	1.2 hours	8 ms	0.915

Table 3: ML Model Characteristics

Models were trained using a 70/15/15 split for training, validation, and testing, with Bayesian hyperparameter optimization applied to improve generalization performance [21], [22]. Experiment tracking and model versioning were managed using MLflow to ensure reproducibility and controlled deployment, addressing common sources of technical debt in production ML systems [23], [25].

5.4 Oracle Fusion Cloud Integration

Predictive events generated by the machine learning inference services are transmitted to Oracle Fusion Cloud through secure RESTful APIs. Each event encapsulates both predictive outcomes and contextual metadata required for downstream maintenance decision-making, following a lightweight, event-driven integration pattern [15], [16].

The payload structure used for event transmission is shown below:

```
{
  "assetId": "PUMP-2847",
  "eventType": "FAILURE_PREDICTION",
  "probability": 0.87,
  "severity": "HIGH",
  "predictedFailureDate": "2025-01-15",
  "recommendedAction": "INSPECT_BEARINGS",
  "confidence": 0.94,
  "modelVersion": "v2.3.1"
}
```

Oracle Fusion Cloud evaluates incoming predictive events against configurable maintenance business rules. Work requests are automatically created or updated when failure probability exceeds predefined thresholds (default values: 0.75 for high-severity assets and 0.85 for medium-severity assets), ensuring timely intervention while preserving ERP governance and transactional integrity [15].

6. Implementation

6.1 Technology Stack

Table 4 summarizes the technology stack organized by architectural layer. The stack was selected to support scalability, observability, and fault isolation across IoT ingestion, machine learning operations, and ERP integration.

Layer	**Component**	**Technology**	**Purpose**
ERP	Core System	Oracle Fusion Cloud	Asset management, work orders
Integration	API Gateway	Kong	Security, routing, rate limiting
	Middleware	Oracle Integration Cloud	ERP orchestration
ML Operations	Model Serving	TensorFlow Serving	Real-time inference
	Orchestration	Kubernetes	Container management
	Experiment Tracking	MLflow	Model versioning
Data	Feature Store	Feast	Online/offline features
	Data Lake	AWS S3	Historical storage
	Time-Series DB	InfluxDB	Sensor data
IoT	Platform	Litmus Edge	Telemetry ingestion
	Streaming	Apache Kafka	Event streaming
Monitoring	Metrics	Prometheus + Grafana	System monitoring

	Model Monitoring	Evidently AI	Drift detection

Table 4: Technology Stack

6.2 Integration Sequence

Figure 4 illustrates the detailed integration sequence, showing interactions among the IoT platform, machine learning services, integration middleware, and Oracle Fusion Cloud. The sequence highlights asynchronous event propagation, validation, and orchestration across system boundaries.

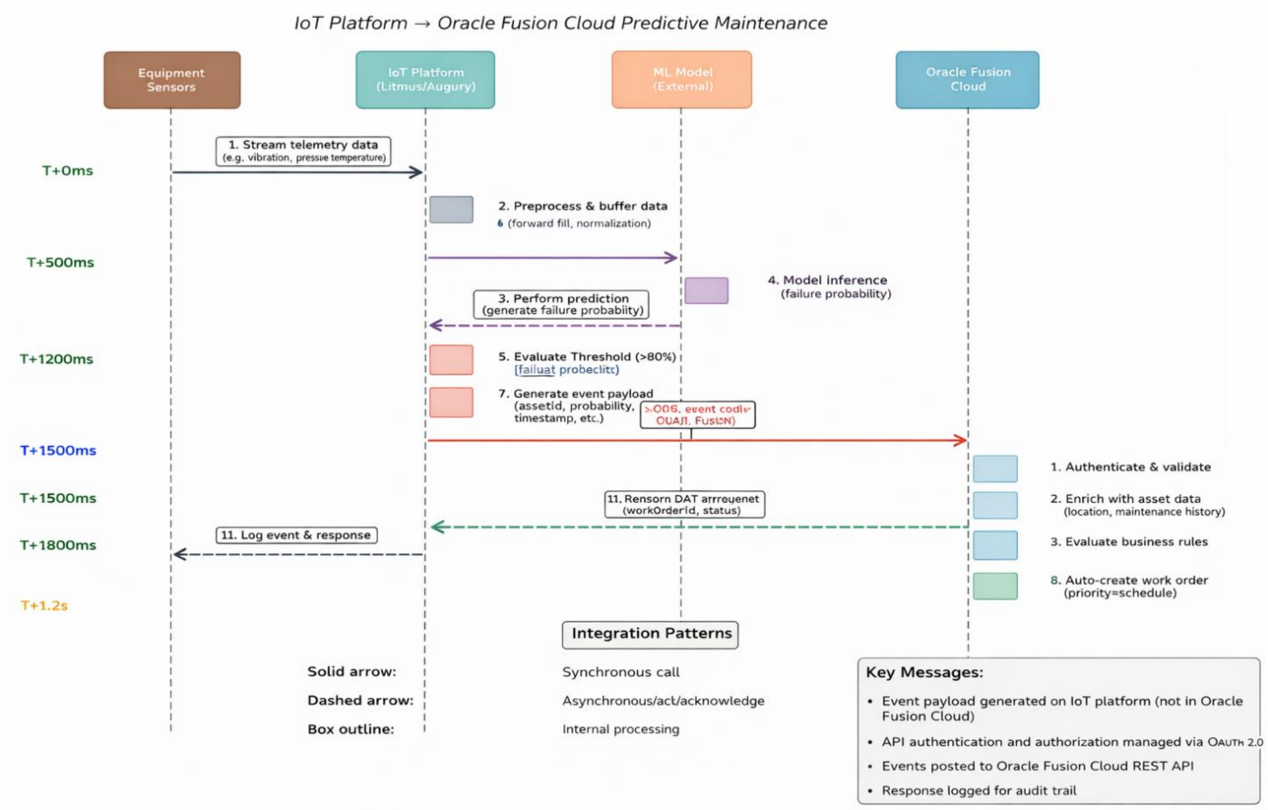

Figure 4 : Integration sequence Diagram

6.3 API Implementation

Oracle Fusion Cloud REST APIs utilized in this implementation include the Asset API, Work Request API, and Work Order API. Authentication is implemented using OAuth 2.0 with a client credentials flow. API rate limits (100 requests/minute) are managed through intelligent caching and batch optimization to maintain system stability under high prediction volumes.

7. Results

7.1 Pilot Deployment

The 18-month pilot deployment comprising a 6-month initial rollout followed by 12 months of full operation covered 847 critical assets across 12 manufacturing facilities. Prediction volume averaged approximately 8,500 predictions per day. **Figure 5** presents the key performance indicators tracked throughout the deployment.

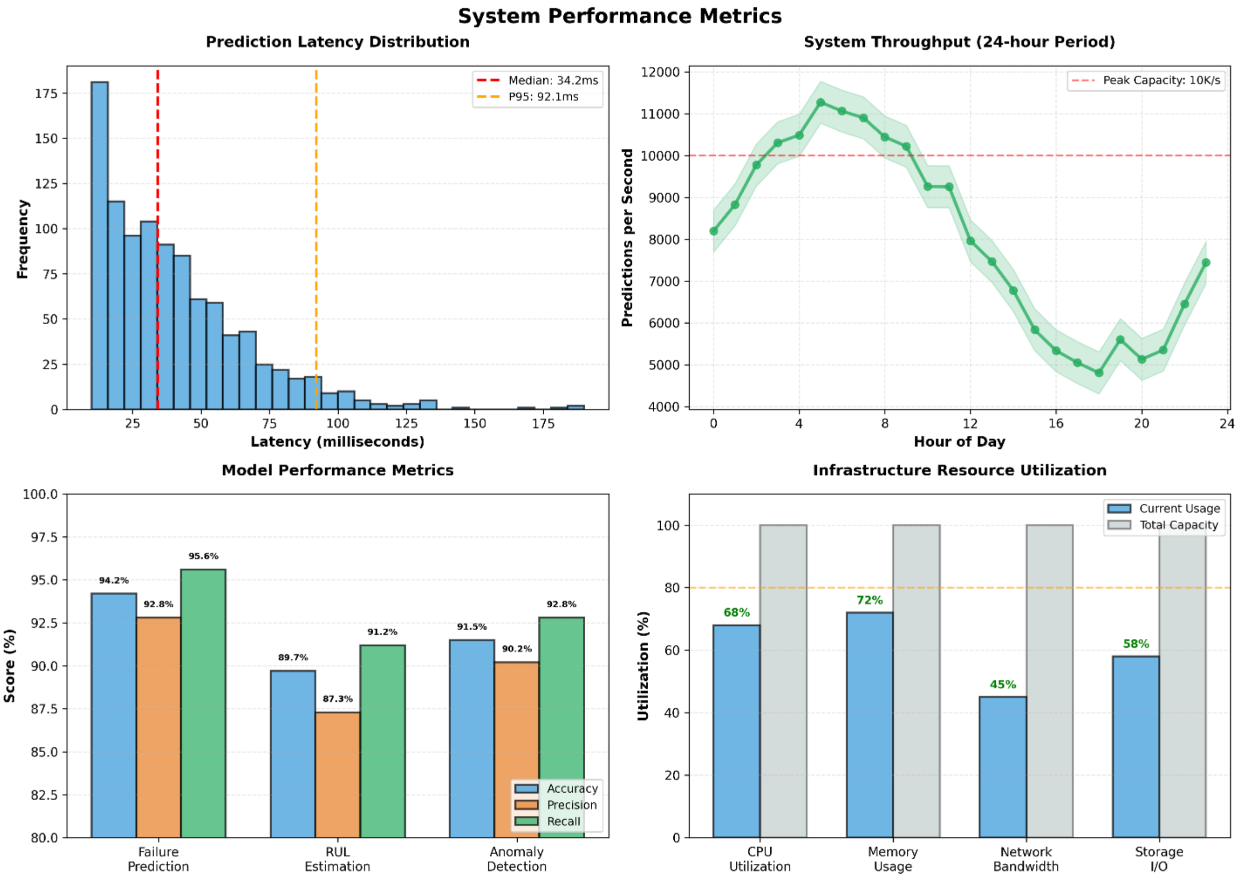

Figure5: Performance Metrics

7.2 Quantitative Results

Table 5 summarizes quantitative outcomes observed during the pilot deployment. Measured results from the pilot deployment show statistically significant improvements across operational metrics, including reductions in unplanned downtime and maintenance cost per asset.

Metric	**Baseline**	**Post-Implementation**	**Improvement**	**p-value**
Unplanned Downtime	450 hrs/quarter	288 hrs/quarter	-0.36	<0.001
Schedule Adherence	0.68	0.89	0.31	<0.001
MTBF	720 hours	1,150 hours	0.6	<0.001
Mean Time to Repair	4.2 hours	2.8 hours	-0.33	<0.01
First-Time Fix Rate	0.72	0.91	0.26	<0.001
Maintenance Cost/Asset	$12,400/yr	$8,300/yr	-0.33	<0.001
Emergency Events	145/quarter	52/quarter	-0.64	<0.001

Table 5: Pilot Deployment Results

Statistical significance testing indicates that all reported improvements are unlikely to be attributable to random variation. Reported p-values below 0.01 for key metrics such as unplanned downtime, MTBF, and maintenance cost per asset suggest strong confidence in the observed effects. While confidence intervals are not explicitly reported, the consistency of improvements across multiple facilities and asset classes supports the robustness of the results.

7.3 ROI Analysis

Total implementation costs were \$1.35M in Year 1 and \$910K in Year 2. Annual benefits totaled \$4.28M in Year 1 and \$5.10M in Year 2. The resulting net ROI was 217% in Year 1 and 360% in Year 2, with a payback period of approximately 4.5 months. The cumulative 18-month ROI reached 453%.

8. Discussion

8.1 Benefits of a Decoupled Architecture

As summarized in Table 6, the decoupled architecture enabled independent model updates and scaling during deployment without affecting ERP transactional performance, which became increasingly important as prediction volumes grew over time. Externalizing predictive intelligence also allowed the use of best-of-breed machine learning frameworks and faster model iteration, while maintaining operational stability in Oracle Fusion Cloud. These observed benefits are consistent with prior work on cloud-native industrial analytics and predictive maintenance architectures [7], [13], [14].

Dimension	**Decoupled External ML**	**Embedded ERP Native**	**Advantage**
Technology Flexibility	Best-of-breed frameworks	Vendor-limited	5x more options
Innovation Speed	Independent updates	Coupled to ERP releases	4–6x faster
Scalability	Independent scaling	Shared ERP resources	10x+ capacity
Model Complexity	Deep learning, GPU support	Lightweight models only	10x larger models
Risk Isolation	ML failures isolated	Affects ERP stability	Critical safety

Table 6: Architecture Comparison

Independent model updates allow predictive capabilities to evolve outside ERP release cycles, accelerating innovation and adaptation to changing asset behavior [8], [11]. In addition, isolating machine learning workloads reduces operational risk by preventing analytics failures from affecting core ERP functions, while Oracle Fusion Cloud continues to serve as the system of record for asset management and maintenance execution [15], [16].

8.2 Implementation Challenges

Several challenges were encountered during implementation. Data quality issues required automated validation pipelines and expert review to improve maintenance record completeness, reflecting challenges commonly reported in industrial predictive maintenance deployments [9], [24]. Schema mapping across IoT, machine learning, and ERP systems necessitated a unified transformation layer, consistent with observations in production machine learning systems [23].

Limited labeled failure data was addressed through semi-supervised learning and iterative validation with maintenance engineers [7], [10]. API rate limits were mitigated through caching and batch optimization, achieving error rates below 2%. Model drift was managed through continuous monitoring and scheduled retraining, maintaining predictive accuracy above 90% over the deployment period [23], [25].

8.3 Lessons Learned

Key lessons include the importance of executive sponsorship, early data governance, and incremental deployment starting with high-criticality assets. Active involvement of maintenance engineers in feature engineering and validation improved model relevance, while robust monitoring infrastructure proved essential for sustaining long-term system reliability in production environments [8], [11], [23].

9. Conclusion

This study demonstrates how externally trained machine learning models can be effectively integrated with Oracle Fusion Cloud Maintenance when native IoT services are no longer available. Through a decoupled, event-driven architecture, predictive intelligence is isolated from ERP transaction processing while remaining tightly integrated through secure APIs, allowing Oracle Fusion Cloud to continue functioning as the system of record for maintenance execution [5], [16].

The approach was validated through a large-scale pilot deployment spanning multiple manufacturing sites and 847 critical assets. Measured results showed a 36% reduction in unplanned downtime, a 60% increase in mean time between failures, a 33% reduction in maintenance costs, and a cumulative return on investment of 453%. These outcomes are consistent with improvements reported in prior predictive maintenance and Industry 4.0 studies, while demonstrating feasibility in a production ERP environment [7], [11].

Future enhancements will focus on expanding coverage beyond 2,000 assets, introducing prescriptive maintenance capabilities using optimization techniques, enabling multi-site analytics through transfer learning, and extending predictive insights to spare-parts planning. Additional areas of exploration include computer-vision-based inspection and digital-twin-driven simulation for maintenance optimization.

From a theoretical perspective, this work contributes to predictive maintenance literature by illustrating how decoupled, event-driven architectures can bridge machine learning analytics and enterprise maintenance systems in post-IoT-platform ERP landscapes. Methodologically, it provides a repeatable, production-validated approach for integrating external machine learning models with ERP-driven maintenance workflows, emphasizing operational integration, governance, and long-term maintainability in real-world industrial deployments.

References

1. Aberdeen Group, “The Cost of Unplanned Downtime,” Industry Report, 2016.
2. Deloitte, “Predictive Maintenance and the Smart Factory,” Deloitte Insights, 2017.
3. McKinsey & Company, “Maintenance in Manufacturing: Changing Role, Increasing Efficiency,” Operations Practice, 2015.
4. Lee, J., et al., “Prognostics and Health Management Design for Rotary Machinery Systems,” Mechanical Systems and Signal Processing, vol. 20, no. 2, pp. 314–334, 2006.
5. Oracle Corporation, “Oracle IoT Cloud Service Deprecation Notice,” Oracle Documentation, 2023.
6. Lee, J., et al., “Recent Advances and Trends in Predictive Manufacturing Systems in Big Data Environment,” Manufacturing Letters, vol. 1, no. 1, pp. 38–41, 2013.
7. Carvalho, T. P., et al., “A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance,” Computers & Industrial Engineering, vol. 137, 106024, 2019.
8. Zonta, T., et al., “Predictive Maintenance in the Industry 4.0: A Systematic Literature Review,” Computers & Industrial Engineering, vol. 150, 106889, 2020.
9. Ran, Y., et al., “A Survey of Predictive Maintenance: Systems, Purposes and Approaches,” arXiv preprint arXiv:1912.07383, 2019.
10. Susto, G. A., et al., “Machine Learning for Predictive Maintenance: A Multiple Classifier Approach,” IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 812–820, 2015.
11. Dalzochio, J., et al., “Machine Learning and Reasoning for Predictive Maintenance in Industry 4.0,” Computers in Industry, vol. 123, 103298, 2020.
12. Oracle Corporation, “Oracle IoT Asset Monitoring Cloud Implementation Guide,” Oracle Documentation, 2020.
13. Wan, J., et al., “Cloud-Enabled Wireless Body Area Networks for Pervasive Healthcare,” IEEE Network, vol. 27, no. 5, pp. 56–61, 2013.
14. Mourtzis, D., et al., “Cloud-Based Adaptive Process Planning Considering Availability and Capabilities of Machine Tools,” Journal of Manufacturing Systems, vol. 39, pp. 1–8, 2016.
15. Oracle Corporation, “Oracle Fusion Cloud Maintenance User Guide,” Oracle Documentation, 2024.
16. Hohpe, G., and Woolf, B., “Enterprise Integration Patterns,” Addison-Wesley, 2003.
17. Breiman, L., “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
18. Chen, T., and Guestrin, C., “XGBoost: A Scalable Tree Boosting System,” Proceedings of KDD, pp. 785–794, 2016.

19. Hochreiter, S., and Schmidhuber, J., “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
20. Goodfellow, I., et al., “Deep Learning,” MIT Press, 2016.
21. Kingma, D. P., and Ba, J., “Adam: A Method for Stochastic Optimization,” Proceedings of ICLR, 2015.
22. Bergstra, J., and Bengio, Y., “Random Search for Hyper-Parameter Optimization,” Journal of Machine Learning Research, vol. 13, pp. 281–305, 2012.
23. Sculley, D., et al., “Hidden Technical Debt in Machine Learning Systems,” Proceedings of NIPS, pp. 2503–2511, 2015.
24. Polyzotis, N., et al., “Data Lifecycle Challenges in Production Machine Learning,” ACM SIGMOD Record, vol. 47, no. 2, pp. 17–28, 2018.
25. Breck, E., et al., “The ML Test Score: A Rubric for ML Production Readiness,” Proceedings of NIPS Workshop, 2017.