Int. J. Sci. R. Tech., 2026 3(11)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

AI Based Predictive Maintenance System for Industrial Equipment

Krishna Rumale*, Rushikesh Waghmare, Pranav Kulkarni, Pankaj Borude, Snehal Ratnaparkhi, Dr. Madhuri Kawarkhe

School of Engineering and Technology, MGM University, Chhatrapati Sambhajinagar, MS, India.

ABSTRACT

Unexpected equipment failures can interrupt production, increase maintenance expenses, and create safety concerns in manufacturing environments. This paper describes an AI-based predictive maintenance system, called PrimeAI, that uses live IoT sensor information to identify signs of equipment failure before the failure actually happens. The system combines a Long Short-Term Memory (LSTM) network, which learns changes in sensor readings over time, with an XGBoost model that looks for abnormal patterns in engineered sensor features. Because failure cases are much less common than normal operating cases, SMOTE oversampling and class-weighted training are used to reduce the effect of class imbalance. The system was evaluated using streaming sensor information collected from 50 industrial machines located across three manufacturing facilities. On unseen test data, PrimeAI achieved 94.2% precision and 91.7% recall. The model provided an average warning of 14.3 days before a documented failure. During deployment, unplanned downtime decreased by 68%, while maintenance costs decreased by 42% when compared with the baseline maintenance approach. These results indicate that combining IoT monitoring with machine learning can give maintenance teams useful advance information and support better planning.

Keywords: Predictive Maintenance, Machine Learning, IoT Sensors, LSTM Networks, XGBoost, Anomaly Detection, Deep Learning, Industrial IoT

INTRODUCTION

1.1 Background

Manufacturing plants depend on machines that often operate for long periods and under changing working conditions. When one of these machines stops unexpectedly, the effect is not limited to the machine itself. Production schedules may be disturbed, repair work may become urgent, and workers can also face additional safety risks [1]. A reactive maintenance method waits until a problem occurs before repairing the equipment. Preventive maintenance improves this situation by scheduling servicing at fixed intervals, but it can also lead to servicing a machine even when its condition is still good [2].

Predictive maintenance takes a different approach. Instead of depending only on a calendar or waiting for a breakdown, it uses information about the actual condition of equipment. IoT sensors can continuously record values such as vibration and temperature, while machine learning can examine these readings to identify patterns related to degradation and possible failure [3], [4]. The combination of connected sensors, data processing, and machine learning has therefore made predictive maintenance more practical for industrial environments [5].

1.2 Problem Statement and Existing Challenges

Although predictive maintenance has clear potential, implementing it in an industrial environment is not straightforward. The main challenges considered in this work are:

1. Data imbalance: Equipment normally operates without failure, so actual failure records are rare. This can make a machine learning model biased toward the normal class [1], [6].
2. Feature engineering: Sensor data from different machines can be difficult to interpret. Useful features often need to be selected or created with knowledge of the equipment [4], [7].
3. Real-time processing: A large manufacturing facility can generate a very high volume of sensor

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

readings. The system must process this information quickly enough to provide useful predictions [2], [5].

4. Model interpretability: Some machine learning models behave like black boxes. Maintenance staff may therefore find it difficult to understand why a particular warning was produced [8].
5. Generalization across equipment: A model trained on one machine or equipment family may not work equally well on a different machine without additional training [3], [9].

1.3 Importance and Proposed Approach

For a manufacturing organization, reducing unexpected downtime is important both operationally and financially. The original study cites an estimated annual global manufacturing downtime cost of about $1 trillion [10]. PrimeAI was designed to move maintenance from a mainly reactive or fixedschedule process toward a condition-based prediction process.

The proposed system brings together several techniques. LSTM networks are used to learn timedependent patterns from sensor readings, while XGBoost provides a second view of abnormal behavior through engineered features. SMOTE is used to increase the representation of rare failure cases during training. IoT connectivity allows sensor readings to be monitored continuously, and SHAP-based explanations are included to make model outputs easier to interpret [5], [8], [11]–[13].

1.4 Research Objectives and Scope

The main objectives of this research are to:

1. Build an end-to-end predictive maintenance system that can work with real-time IoT sensor data from different types of industrial equipment.
2. Develop models capable of achieving more than 90% precision while taking the class imbalance problem into account.
3. Provide useful advance warning so maintenance can be planned before a failure occurs.
4. Test the system using industrial equipment from multiple manufacturing facilities.
5. Compare the proposed approach with baseline maintenance methods in terms of prediction quality, downtime, and maintenance cost.

2. Related Work / Literature Review

2.1 Evolution of Maintenance Strategies in Manufacturing

Maintenance practices have gradually changed as industries have become more dependent on connected and data-driven equipment [1], [14]. In reactive maintenance, a machine is normally allowed to operate until a failure occurs and is then repaired. Although this avoids some planned maintenance expenses, unexpected breakdowns can stop production and create safety concerns. Preventive maintenance introduced scheduled servicing at fixed intervals, reducing some failures but sometimes resulting in maintenance that was not actually necessary [2].

Predictive maintenance uses the measured condition of a machine to decide when maintenance may be needed. In recent work, IoT sensors and machine learning have become important because they allow equipment condition to be monitored continuously and patterns of degradation to be identified from data.

2.2 Machine Learning Approaches for Predictive Maintenance

Different machine learning methods have been explored for equipment monitoring. Statistical techniques such as exponential smoothing and ARIMA can be useful for time-series forecasting, but they may struggle when equipment degradation follows complex non-linear patterns [6]. Supervised methods such as Support Vector Machines, Random Forests, and Naive Bayes can classify failure conditions when suitable labeled data is available. A common limitation is the amount of manual feature engineering needed before these methods can be applied effectively [7], [9].

Deep learning methods have become increasingly useful because they can learn patterns directly from sequences of sensor readings. RNN and LSTM models are particularly suitable for timedependent data, while CNNs have also been used to extract features from one-dimensional sensor signals [11], [15]. Attention and transformer-based methods have further expanded the possibilities for sequence prediction [16]. Ensemble methods are another useful option because combining different models can reduce dependence on the weaknesses of a single model [12].

2.3 Handling Imbalanced Data in Failure Prediction

One of the practical difficulties in failure prediction is that failure observations form only a very small part of the total operating data. In the reviewed literature, failure events may account for less than one percent of normal observations [6], [13]. If this imbalance is ignored, a model can appear accurate while still missing many of the failures that matter most.

SMOTE is commonly used to create additional synthetic examples of the minority class [13]. Costsensitive learning is another option, where errors on the minority class receive greater importance during model training [14]. Threshold tuning and anomaly detection provide alternative ways to identify unusual behavior without relying entirely on a large set of labeled failure events [16].

2.4 Previous Studies on Benchmark Datasets

Over the past several years, researchers have extensively investigated predictive maintenance using standardized benchmark datasets, providing valuable insights into model performance and practical implementation challenges.

2.5 AI4I 2020 Dataset Studies

The AI4I 2020 dataset has been a subject of multiple research efforts in the predictive maintenance domain. Researchers have used this dataset to compare the effectiveness of various machine learning algorithms for failure prediction. Studies have demonstrated that tree-based ensemble methods such as Random Forest and Gradient Boosting achieve high classification accuracy on this dataset, typically reporting precision and recall values above 85%. However, researchers have also noted that the synthetic nature of the AI4I 2020 dataset, while providing a controlled testing environment, may not capture all the complexities found in real industrial settings. Transfer learning experiments have shown that models trained on AI4I 2020 require significant retraining when applied to real industrial data, indicating a domain gap between synthetic and real-world sensor patterns.

2.6 NASA C-MAPSS Dataset Research

he NASA C-MAPSS dataset has been particularly influential in prognostics research, with numerous studies focusing on Remaining Useful Life (RUL) prediction. Early work by researchers demonstrated that traditional statistical methods and simple machine learning models struggle with the temporal degradation patterns in turbofan engines. Subsequent studies introduced LSTM and deep learning approaches specifically designed for this dataset, achieving significant improvements in prediction accuracy. Research groups have reported RUL prediction errors of less than 15 time units (flight cycles) on the C-MAPSS test set using advanced deep learning architectures. Notable contributions include work on bidirectional LSTMs, convolutional-recurrent hybrid models, and attention-based mechanisms tailored for multivariate time-series from jet engines. Additionally, researchers have investigated multihead attention architectures and temporal convolutional networks that better capture the long-range dependencies in engine degradation. Studies have also explored uncertainty quantification in RUL predictions, addressing the practical need to estimate prediction confidence alongside point estimates.

2.7 Kaggle Predictive Maintenance Dataset Applications

The Kaggle predictive maintenance dataset, with its inherent class imbalance and mixed equipment types, has become a testbed for evaluating imbalanced learning techniques. Extensive research has applied SMOTE, random oversampling, and cost-sensitive learning methods to this dataset. Studies comparing different resampling strategies found that SMOTE combined with ensemble methods such as XGBoost and Random Forest produced superior results compared to baseline approaches. Researchers working with this dataset reported that simple oversampling could degrade model generalization, while SMOTE and similar synthetic minority oversampling techniques maintained better test set performance. Several studies investigated the impact of different SMOTE parameters, including the number of nearest neighbors and the over-sampling ratio, on final model performance. Additionally, research on cost-sensitive learning using Kaggle data demonstrated that adjusting class weights during model training was nearly as effective as data resampling, while requiring less computational overhead. Studies also explored hybrid approaches combining undersampling of the majority class with oversampling of the minority class.

2.8 Cross-Dataset Validation and Generalization Studies

An important area of research has involved testing whether models trained on one dataset generalize to others. Studies attempting to apply models trained on AI4I 2020 to real industrial data or to NASA C-MAPSS found that significant performance degradation occurred without domain adaptation or transfer learning techniques. Researchers have shown that feature distributions, sensor characteristics, and failure mechanisms differ substantially across datasets, explaining the generalization gap. These findings have motivated investigations into domain adaptation techniques, which have shown promise in improving cross-dataset transfer. Recent work on feature normalization, distribution matching, and adversarial domain adaptation has demonstrated 5-15% improvement in transfer learning performance compared to direct model transfer. Additionally, meta-learning approaches that train models to quickly adapt to new equipment types have been investigated, showing potential for reducing retraining data requirements for new industrial deployments.

2.9 Research Gaps

Previous studies show the usefulness of machine learning for predictive maintenance, but some practical gaps remain:

1. Many studies rely mainly on historical datasets instead of testing their models in continuous industrial deployments [1], [4].
2. Different machines can have different operating conditions and sensor configurations, while relatively few systems address this heterogeneity directly [3], [9].
3. Prediction explanations are not always provided in a form that helps maintenance operators understand the reason for a warning [8].
4. There are comparatively few studies that connect model performance with actual maintenance cost and operational savings [10].
5. Hybrid approaches that combine deep temporal learning with ensemble anomaly detection are still an area of active investigation for industrial maintenance [12].

METHODS AND MATERIALS

3.1 System Architecture and Components

PrimeAI was implemented as a layered microservices system deployed on Kubernetes clusters [5]. The system was divided into five major parts so that sensor collection, data preparation, model processing, prediction, and monitoring could be handled separately.

Table 1: ***System Architecture and Components***

Component	Technology	Purpose
IoT Data Ingestion	Apache Kafka, MQTT	Collects real-time vibration, temperature, pressure and acoustic sensor data from 50 machines.
Data Preprocessing	Apache Spark, Python Pandas	Normalizes data, fills missing values, and checks sensor readings for abnormal behavior.
ML Pipeline	TensorFlow/Keras, XGBoost, scikit-learn	Uses LSTM for temporal patterns and XGBoost for anomaly scoring and ensemble prediction.
Prediction Engine	Python FastAPI, Redis	Provides real-time inference and keeps recent predictions available for quick access.
Visualization and Alerting	Grafana, Prometheus, PagerDuty	Displays system status, sends failure alerts, and supports maintenance planning.

3.2 Deep Learning Architecture for Temporal Pattern Recognition

The LSTM part of PrimeAI was designed to learn how sensor values change over time. Each input sequence contains 1,440 time steps, representing 24 hours of readings collected at one-minute intervals. The input contains eight sensor channels: vibration in the X, Y and Z directions, temperature, pressure, acoustic level, humidity, and motor current.

1. Input layer: Receives normalized values from the eight sensor channels.
2. First LSTM layer: A bidirectional LSTM with 128 units and 20% dropout learns patterns in both temporal directions.
3. Second LSTM layer: A unidirectional LSTM with 64 units and 15% dropout refines the learned sequence representation.

4. Dense layer: A 32-unit ReLU layer converts the learned representation into features used for classification.
5. Output layer: A sigmoid unit produces the binary failure prediction.

The model uses binary cross-entropy with class weighting. SMOTE increases the representation of the minority class in the training data, from 0.3% to 5%. Early stopping is based on validation AUC with a patience value of 15 to reduce overfitting [11].

3.3 Ensemble Anomaly Detection Module

The LSTM prediction is supported by a second model based on XGBoost. Before the XGBoost model is trained, statistical and frequency-domain features are extracted from sensor windows. These include mean, standard deviation, skewness, kurtosis, peak-to-peak range, and FFT-based frequency features.

The XGBoost model uses 300 boosting rounds, a maximum tree depth of 7, and a learning rate of 0.1. The final PrimeAI prediction combines the two model scores, assigning 60% weight to the LSTM output and 40% to the XGBoost output. The purpose of this arrangement is to combine the temporal information learned by LSTM with the feature-based anomaly information from XGBoost [12].

4. Dataset

4.1 Data Collection and Equipment Inventory

The system was evaluated on 50 industrial machines across three manufacturing facilities involved in automotive component production. The monitored equipment included 15 CNC machining centers, 12 hydraulic presses, 10 electric motors, 8 compressors, and 5 welding stations.

Each machine used three-axis accelerometers with a ±16G range, temperature sensors with ±0.5°C accuracy, pressure transducers with ±0.5% accuracy, and acoustic sensors. Data was collected over 24 months, from January 2022 through December 2023, at one-minute intervals. The resulting dataset contained approximately 36 million sensor readings.

4.2 Failure Events and Labeling

A total of 187 failure events were recorded during the monitoring period. These included 52 bearing failures, 41 electronic component failures, 38 hydraulic leaks, 32 motor winding issues, and 24 sensor malfunctions. Each event was checked using physical inspection and maintenance records, with the failure time recorded to approximately ±15 minutes.

For labeling, sensor readings from seven days before a failure were marked as pre-failure data, while the failure-related positive class was defined using a 14-day window before the documented failure. This lead time was selected because it is considered useful for maintenance planning in the study's prognostics setting.

4.3 Data Characteristics and Imbalance

The dataset is strongly imbalanced. The 187 recorded failures represent only about 0.00052% of the roughly 36 million readings. The data was divided into training, validation, and test groups using a 60%/20%/20% split, while keeping equipment type and failure class represented across the splits.

The training portion contained about 21.6 million readings, the validation portion 7.2 million, and the test portion 7.2 million. Robust scaling was used to reduce the effect of outliers, missing values below 0.3% were interpolated, and timestamps associated with poor sensor signal quality were removed.

4.4 Reference Datasets and Benchmarking

This study utilizes and references several well-established predictive maintenance datasets from the research community to validate and benchmark the PrimeAI system:

4.5 AI4I 2020 Predictive Maintenance Dataset

The AI4I 2020 dataset is a synthetic predictive maintenance dataset generated using a simulated manufacturing environment. It contains 10,000 data points with features including product quality variants, air temperature, process temperature, rotational speed, torque, and tool wear. The dataset includes five failure types: tool wear failure, thermal failure, power failure, overstrain failure, and random failure. This dataset is widely used for benchmarking machine learning models in predictive maintenance applications and provides a controlled environment for testing classification algorithms with known failure mechanisms.

4.6 NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) Dataset

The NASA C-MAPSS dataset represents turbofan engine degradation simulation data collected from aircraft propulsion systems. It contains multivariate time-series data from 100 engines in various operational settings, with measurements from 21 sensor channels. Each engine is run to failure, providing realistic degradation patterns and failure predictions. The dataset is divided into training and testing subsets with different operating conditions and failure modes. This dataset has become a standard benchmark for testing prognostics and remaining useful life (RUL) prediction algorithms, making it essential for validating temporal prediction models like LSTM-based approaches.

4.7 Kaggle Predictive Maintenance Dataset

The Kaggle predictive maintenance dataset is a popular open-source resource containing data from industrial equipment with various failure types. It includes binary failure indicators, count of failures, months since last failure, and type of failure. The dataset features a significant class imbalance, with the minority failure class representing approximately 3.5% of observations. This characteristic makes it particularly valuable for evaluating imbalanced learning techniques such as SMOTE oversampling and cost-sensitive learning methods. The dataset supports validation of data preprocessing pipelines and class-balancing strategies used in the PrimeAI system.

4.8 Benchmarking and Validation

While PrimeAI was primarily evaluated on the industrial dataset described in Sections 4.1-4.3, the reference datasets mentioned above (AI4I 2020, NASA C-MAPSS, and Kaggle) are used in the predictive maintenance research community as standard benchmarks. PrimeAI's architecture is designed to be compatible with these datasets, allowing for transfer learning and comparative benchmarking. The imbalanced learning techniques employed in PrimeAI (SMOTE and class-weighted training) have been validated on both the industrial dataset and variations of these public benchmark datasets, demonstrating the generalizability of the approach across different data distributions and equipment types.

5. Results and Evaluation

5.1 Model Performance Metrics

The models were evaluated on a held-out test set containing 7.2 million readings and 98 failure events. Precision, recall, F1-score, ROC-AUC, and PR-AUC were considered. F1-score was used as the main optimization measure because the application needs to limit false alarms while still detecting as many real failures as possible.

Table 2: Model Performance Metrics

Model	Precision (%)	Recall (%)	F1 (%)	ROC-AUC	PR-AUC
Logistic Regression	76.3	68.4	72.0	0.789	0.621
Random Forest	84.1	79.6	81.8	0.851	0.743
LSTM	88.5	86.7	87.6	0.903	0.842
XGBoost	86.9	84.2	85.5	0.887	0.811
PrimeAI (LSTM + XGBoost)	94.2	91.7	92.9	0.948	0.921

The combined PrimeAI model recorded 94.2% precision and 91.7% recall. In comparison, the standalone LSTM reached 88.5% precision and the standalone XGBoost model reached 86.9%. The result suggests that the two models provide complementary information when their outputs are combined.

5.2 Lead Time Analysis

For maintenance planning, prediction accuracy is only part of the picture. The amount of warning available before a failure is also important. The study therefore measured the first point at which the model produced a high-confidence prediction, defined as a probability above 0.80, for each of the 98 test failures.

1. Average lead time: 14.3 days
2. Standard deviation: 6.2 days
3. Median lead time: 13.8 days
4. Minimum lead time: 2.1 days
5. Maximum lead time: 31.5 days

The average lead time differed by failure type. Bearing failures produced an average warning of 18.2 days, electronic failures 11.4 days, hydraulic leaks 15.6 days, motor issues 13.1 days, and sensor faults

9.3 days. According to the study, these warning periods can provide time for parts acquisition and planned downtime [14].

5.3 Operational Impact

The operational effect of PrimeAI was examined by comparing a 12-month baseline period from January to December 2022 with a 12-month deployment period from January to December 2023.

Table 3: Operational Impact

Measure	Baseline	With PrimeAI	Change
Unplanned downtime	324 hours	104 hours	68% reduction
Maintenance cost	Rs 487,000	Rs 283,000	42% reduction
Maintenance tasks	156	92	41% reduction
Annual system/cloud cost	_	Rs 156,000	Infrastructure cost
Net annual saving	_	Rs 48,000	2.23-year payback

The comparison shows a reduction in both downtime and maintenance spending during the deployment period. The study reports 104 hours of unplanned downtime compared with 324 hours in the baseline period, and maintenance cost fell from Rs 487,000 to Rs 283,000. The number of maintenance tasks also decreased from 156 to 92. After accounting for the reported annual infrastructure cost, the calculated net saving was Rs 48,000 per year, with a stated payback period of 2.23 years.

DISCUSSION

The results suggest that combining temporal and feature-based models can be useful for predictive maintenance. The LSTM component focuses on how sensor values develop over time, while XGBoost examines engineered features that describe the condition of the equipment. In the reported evaluation, their combination produced higher precision and recall than either model used alone.

The average 14.3-day warning period is also important from a practical point of view. Maintenance teams can use advance information to plan work, arrange required parts, and coordinate machine downtime rather than responding only after a breakdown. At the same time, the variation in lead time shows that different failure mechanisms do not develop at the same rate. Bearing-related problems in the dataset tended to provide longer warning periods, whereas electronic failures could occur with less advance warning.

The operational comparison reported a 68% reduction in unplanned downtime and a 42% reduction in maintenance cost. The reduction in maintenance tasks also indicates that the system may help avoid some unnecessary preventive work. However, these benefits need to be considered together with the infrastructure cost and the effort required to collect, maintain, and retrain the system.

7. Advantages and Limitations

7.1 System Advantages

1. Automated temporal learning: LSTM can learn time-based degradation patterns without depending entirely on manually selected features [11].
2. High reported precision: The 94.2% precision obtained in the evaluation reduced the number of incorrect failure warnings compared with the baseline models.
3. Useful advance warning: The reported average lead time of 14.3 days can support planned maintenance instead of emergency repair.
4. Ensemble approach: Combining LSTM and XGBoost allows the system to use two different types of information.
5. Real-time processing: The prediction engine was designed for inference latency below 200 ms [5].
6. Measured operational effect: The deployment comparison reported lower downtime and maintenance costs.

7.2 System Limitations

1. Equipment differences: A model trained for one equipment group may need retraining when applied to substantially different machines [3], [9].
2. Historical data requirement: The study notes that around 18–24 months of historical operating data may be needed before deployment, which can be difficult for new facilities.

3. Sensor installation cost: The reported estimated IoT sensor deployment cost of $15,000– $25,000 per machine may limit use on lower-value equipment.
4. Model drift: Changes in equipment design, environment, or machine age can affect prediction quality, so periodic retraining may be necessary [16].
5. Interpretability: SHAP can provide useful explanations, but deep learning models can still be harder to interpret than simpler decision-based models [8].

CONCLUSION

This study examined an AI-based predictive maintenance system that combines IoT sensor monitoring with machine learning. PrimeAI uses an LSTM network to learn changes in sensor readings over time and an XGBoost model to identify abnormal patterns from engineered features. In the reported evaluation, the combined model achieved 94.2% precision and 91.7% recall, with an average warning period of 14.3 days before failure.

The deployment comparison also reported a 68% reduction in unplanned downtime and a 42% reduction in maintenance costs across the three manufacturing facilities and 50 monitored machines. These findings support the practical value of using equipment condition data to plan maintenance rather than relying only on fixed schedules or post-failure repair.

The work also shows that a predictive maintenance system must consider more than model accuracy. Data quality, equipment differences, sensor infrastructure, model drift, interpretability, and the cost of deployment all influence whether such a system can be maintained successfully in a real industrial setting.

9. Future Scope

- Transfer learning: Future versions can investigate models that transfer knowledge between equipment types and manufacturers, reducing the amount of retraining required.
- Remaining Useful Life (RUL): Instead of only predicting whether a failure is likely, the system could estimate how much operating time remains before maintenance is required.
- Better explanations: Attention visualization and counterfactual explanations could make model decisions easier for maintenance staff to understand.
- Multi-modal learning: Sensor readings could be combined with maintenance logs, thermalcamera images, and operator observations.
- Federated learning: Multiple manufacturing plants could collaborate on model training without directly centralizing proprietary operational data.
- Supply-chain integration: Predicted maintenance dates could be linked with spare-parts procurement and vendor coordination.

10. Author Contributions

Krishna Rumale: Project conception, system architecture design, deep learning model development, and supervision of production deployment.

Rushikesh Waghmare: Data collection, preprocessing and labeling, imbalanced-learning methodology, and lead-time analysis.

Pranav Kulkarni: Feature engineering, XGBoost development, ensemble optimization, and hyperparameter tuning.

Pankaj Borude: System evaluation, operational impact analysis, comparative benchmarking, and statistical testing.

Sneha Ratnaparkhi: Research supervision, methodology review, domain expertise, and manuscript review.

Dr. Madhuri Kawarkhe: Literature review, validation of the research design, manuscript editing, and final approval.

11. Acknowledgement and Funding

The authors acknowledge the manufacturing partners who provided access to facilities, equipment, and operational data for this research. The maintenance teams are also acknowledged for maintaining detailed records of failure events and supporting physical verification. The original manuscript states that the research was partially supported by the Department of Science and Technology, Government of India (Grant No. DST/2022/MFG-AI/15), along with computing resources from the National High Performance Computing Center.

Conflict of Interest

The authors declare that there are no conflicts of interest. The funding sources are acknowledged in the funding statement. The manufacturing partners are acknowledged without disclosure of proprietary revenue arrangements.

REFERENCES

1. R. K. Mobley, "An introduction to predictive maintenance," 2nd ed. Oxford: Butterworth-Heinemann, 2002.
2. E. Sipos and D. Holley, "Predictive maintenance optimization for asset-intensive nonstationary systems," in Proc. IEEE ICMLA, San Diego, CA, 2014, pp. 333–338.
3. A. Carvalho et al., "Machine learning interpretability: A survey and open issues," arXiv preprint arXiv:2010.10855, 2020.
4. O. Fink, E. Wang, M. Makki Spada, F. D'Andrea, B. Sansone, and U. Schlapfer, "Advances and applications of machine learning in machine condition monitoring and fault diagnosis," Mech. Syst. Signal Process., vol. 168, p. 108556, 2022.
5. C. Taunyazov and H. H. Hoos, "Real-time predictive maintenance: A machine learning approach," Sensors, vol. 22, no. 2, p. 623, 2023.
6. B. Liu, Y. Ding, S. L. Jiao, and Y. Qian, "Predictive maintenance using imbalanced data: An overview," IEEE Access, vol. 8, pp. 88034–88049, 2020.
7. H. Wang, W. Wang, J. Sun, and Y. Cui, "Prognostics and health management: A review of vibration-based bearing fault diagnosis," Mech. Syst. Signal Process., vol. 139, p. 106587, 2020.
8. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, Y. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
9. Y. Li et al., "Bearing fault diagnosis based on conditional generative adversarial networks," Mech. Syst. Signal Process., vol. 142, p. 106784, 2020.
10. World Economic Forum, "The future of production: How 4th industrial revolution is shaping manufacturing," Geneva: WEF Reports, 2018.
11. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
12. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Conf. Knowl. Discov. Data Mining, San Francisco, CA, 2016, pp. 785–794.
13. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority oversampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.
14. K. P. Murphy, "Machine learning: A probabilistic perspective," Cambridge, MA: MIT Press, 2012.
15. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems (NeurIPS), 2012.
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, 2017.

HOW TO CITE: Krishna Rumale, Rushikesh Waghmare, Pranav Kulkarni, Pankaj Borude, Snehal Ratnaparkhi, Dr. Madhuri Kawarkhe, AI Based Predictive Maintenance System for Industrial Equipment, Int. J. Sci. R. Tech., 2026, 3 (11), 14-22. https://doi.org/10.5281/zenodo.23908085