Predictive Maintenance for Turbofan Jet Engines Using Machine Learning and Time-Series Modeling

Mamadou Ndiong — University of Maryland — mndiong@umd.com

***Abstract*—Predictive maintenance plays a vital role in minimizing unexpected machinery failures and optimizing maintenance schedules in critical industries such as aerospace and manufacturing. This study investigates the application of machine learning and deep learning techniques for Remaining Useful Life (RUL) prediction using the NASA C-MAPSS turbofan engine degradation dataset. We develop a comprehensive data preprocessing and feature engineering pipeline, including sensor filtering, normalization, rolling statistics, and delta features. Additionally, we introduce degradation-phase training, which focuses model learning on the latter half of each engine's operational life where degradation is most prominent. Experimental results show that XGBoost, trained exclusively on degradation data, achieves significant performance gains, outperforming both full-lifespan training and deep learning models. The best model achieved a mean absolute error (MAE) of 17.7 cycles and root mean squared error (RMSE) of 25.07 cycles on FD003. These findings provide practical guidance for scalable deployment of predictive maintenance systems in industrial settings.**

I. Introduction

Unplanned equipment failures in critical industries such as aerospace, energy, and manufacturing can result in catastrophic financial losses, safety hazards, and operational disruptions. Traditional maintenance strategies, including preventive and reactive maintenance, either fail to prevent breakdowns or incur unnecessary downtime and costs. Predictive maintenance, powered by machine learning and sensor data, offers a promising solution by forecasting failures before they occur, enabling proactive intervention and optimal resource allocation.

Prognostics and health management (PHM) has emerged as a key area of research aimed at developing algorithms that can predict the Remaining Useful Life (RUL) of machinery components. Accurate RUL estimation is vital for reducing maintenance costs, increasing asset availability, and extending system lifespan. However, the task is highly challenging due to the complex, nonlinear, and multivariate nature of degradation processes under varying operational conditions.

The NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset has become a benchmark for developing and evaluating predictive maintenance algorithms. It provides simulated degradation trajectories of turbofan engines under multiple operating conditions and fault modes. This study focuses on leveraging the **FD001**, **FD002**, and **FD003** subsets of C-MAPSS to evaluate the performance of three machine learning approaches: Random Forest, XGBoost, and Long Short-Term Memory (LSTM) networks.

The primary contributions of this work are:

1) A comprehensive feature engineering pipeline including scaling, rolling statistics, and delta features to enhance degradation signal representation.
2) A comparative analysis of traditional tree-based models (Random Forest, XGBoost) and deep learning approaches (LSTM) for RUL prediction.
3) Practical recommendations for the deployment of scalable predictive maintenance solutions based on experimental results.

This paper is organized as follows: Section II reviews related work; Section III describes the data and preprocessing methods; Section IV presents the modeling approaches; Section V discusses the results; Section VI provides analysis and recommendations; and Section VII and VIII conclude with limitation suggestions for future work.

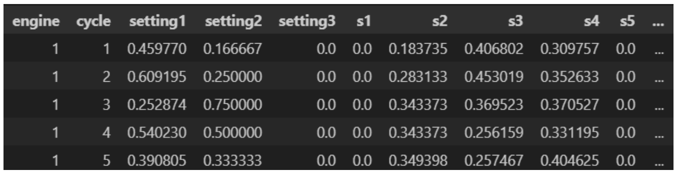
II. Limitations

While this study demonstrates effective approaches for RUL prediction using NASA C-MAPSS data, several limitations should be noted:

- **Synthetic dataset:** The NASA dataset is simulated. Real-world engines exhibit more noise, sensor drift, and irregular sampling.
- **Limited model tuning:** LSTM models were not extensively tuned due to compute constraints. Hyperparameter optimization or hybrid architectures could yield better results.
- **No fault type classification:** The models only predict time-to-failure but do not identify the failure mode, which may be important for some applications.
- **Static deployment assumptions:** We did not test live-streaming scenarios or edge deployment, which are essential in aviation or IoT contexts.

III. Related Work

Predictive maintenance has seen advancements through data mining and deep learning. Early work used survival models and decision trees; recent studies have explored XGBoost and LSTM architectures for time-series degradation analysis. The C-MAPSS dataset remains a standard for benchmarking such models. Michael T. Tong from the Glenn Research Center in Cleveland, Ohio has also made use of the full dataset and published the paper **Using Machine Learning to Predict Core Sizes of High-Efficiency Turbofan Engines**



engine	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...
1	1	0.459770	0.166667	0.0	0.0	0.183735	0.406802	0.309757	0.0	...
1	2	0.609195	0.250000	0.0	0.0	0.283133	0.453019	0.352633	0.0	...
1	3	0.252874	0.750000	0.0	0.0	0.343373	0.369523	0.370527	0.0	...
1	4	0.540230	0.500000	0.0	0.0	0.343373	0.256159	0.331195	0.0	...
1	5	0.390805	0.333333	0.0	0.0	0.349398	0.257467	0.404625	0.0	...

Fig. 1. First 5 rows of dataset after applying column names

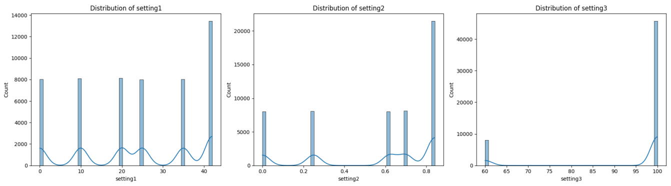


Fig. 2. Enter Caption

IV. Methods

A. Data Description

The dataset used in this study is the NASA C-MAPSS Turbofan Engine Degradation Simulation dataset, widely recognized as a benchmark for prognostics research. It contains multivariate time-series data of engine performance recorded over the full operational lifetime of simulated jet engines. Each data record corresponds to a single engine at a specific time cycle and includes three operational settings and twenty-one sensor measurements. These measurements capture variables such as fan speed, temperatures, pressure ratios, and vibration levels, which reflect the health state of the engine over time. The dataset is divided into four subsets; this study focuses on the following three:

FD001: Consists of 100 training engines and 100 test engines. All engines operate under a single operating condition and exhibit a single fault mode.

FD002: Contains 260 training engines and 259 test engines. This set introduces greater complexity with six distinct operating conditions while maintaining a single fault mode.

FD003: Comprises 100 training engines and 100 test engines. Engines operate under a single condition but can experience multiple fault modes simultaneously, increasing the difficulty of accurate RUL estimation.

The objective is to predict the Remaining Useful Life (RUL) of each engine, defined as the number of cycles remaining before system failure. The dataset provides full run-to-failure trajectories for the training engines and partial trajectories for the test engines, with corresponding ground truth RUL labels provided separately.

This structure allows for testing predictive maintenance models in a controlled yet realistically challenging environment, simulating how models would operate on assets in live industrial settings.

B. Preprocessing

To ensure data consistency and optimize model performance, a comprehensive preprocessing pipeline was applied to all datasets. The preprocessing steps are summarized as follows:

To ensure data consistency and optimize model performance, a comprehensive preprocessing pipeline was applied to all datasets. The preprocessing steps are summarized as follows:

1) **Column Renaming:** The raw datasets were provided without meaningful column labels. We assigned clear column names for all features, including the operational settings (setting1, setting2, setting3) and twenty-one sensor measurements (s1 to s21).
2) **Variance-Based Feature Selection:** An initial exploratory analysis revealed that some sensor features, such as s16, exhibited near-zero variance across all engine cycles. These features were determined to be uninformative and were removed to reduce dimensionality and potential noise.
3) **Feature Normalization:** All continuous variables, including operational settings and sensor readings, were scaled using MinMaxScaler to a range of [0, 1]. Normalization is critical to ensure that all features contribute equally during model training, preventing high-magnitude features from dominating model learning.
4) **Cycle Normalization:** To incorporate degradation context, a normalized cycle life feature was added for each record. This feature represents the ratio of the current cycle to the maximum cycle number for the engine ($cycle\_norm = cycle/max\_cycle\_engine$), providing a relative measure of progression through the engine's operational life.
5) **Feature Engineering – Rolling Statistics:** Time-series trends are often more indicative of degradation than raw values. For each key sensor (s2, s3, s4, s7, s9, s11, s12, s14, s17, s20, s21), we computed rolling mean and rolling standard deviation over a moving window of five cycles. This captures both the local trend and variability, which are critical for anticipating failure.
6) **Feature Engineering – Delta Features:** To capture short-term changes that may signal emerging faults, we calculated delta features for the same set of key sensors. The delta is defined as the difference in the sensor reading between the current cycle and the previous cycle ($\Delta = current\_value - previous\_value$).
7) **Target Generation for Training:** For the training dataset, Remaining Useful Life (RUL) values were calculated for each time step as the difference between the engine's final cycle and the current cycle. This provided the ground truth target for supervised learning models.

By applying these systematic preprocessing and feature engineering steps, we enhanced the information content of the dataset and improved the model's ability to detect subtle degradation patterns, which are essential for accurate RUL prediction.

C. Modeling

We evaluated three machine learning approaches for Remaining Useful Life (RUL) prediction: Random Forest Regressor, XGBoost Regressor, and Long Short-Term Memory

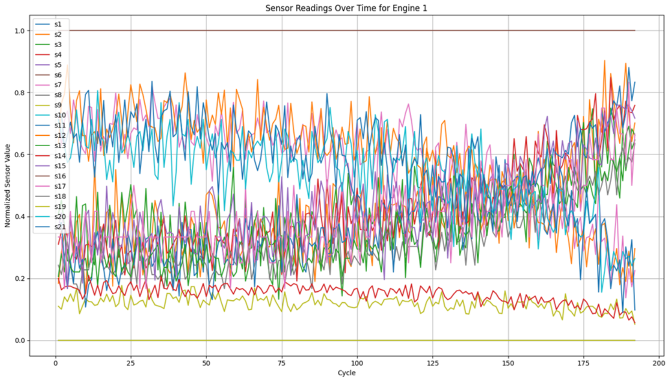


Fig. 3. Sensors readings of Engine 1

(LSTM) networks. Each approach used the preprocessed and feature-engineered datasets described previously.

1) Random Forest Regressor: Random Forest is an ensemble learning method that builds multiple decision trees and outputs the average prediction of individual trees. It is robust to noise and overfitting, especially with tabular data, making it a strong baseline model. The Random Forest model was trained using the engineered tabular dataset without sequential dependencies.

2) XGBoost Regressor: Extreme Gradient Boosting (XGBoost) is an optimized implementation of gradient boosting decision trees. XGBoost has been widely used for structured data problems due to its superior speed, regularization capabilities, and accuracy. As with Random Forest, XGBoost was trained using the tabular feature set but often demonstrated improved performance due to its boosting framework.

3) Long Short-Term Memory (LSTM) Networks: LSTM networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies in sequential data. LSTM models were trained using sliding windows of 30 consecutive cycles from each engine, with the model predicting RUL at the last time step of each window. This approach allows the model to capture temporal patterns of degradation that may be missed by tree-based models.

4) Model Evaluation: All models were evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics on the test set. For Random Forest and XGBoost, predictions were made on the last available cycle of each test engine. For LSTM, sequences of the final 30 cycles were used for predictions. The evaluation provided a consistent basis for comparing traditional machine learning models with deep learning approaches.

V. Results

A. FD002 Results

We evaluated models on FD002 using both full-life training and degradation-phase training. The latter involved restricting the training data to only the final 50% of each engine's life cycle to better capture degradation patterns. As shown in Table I, degradation-phase training significantly improved model performance.

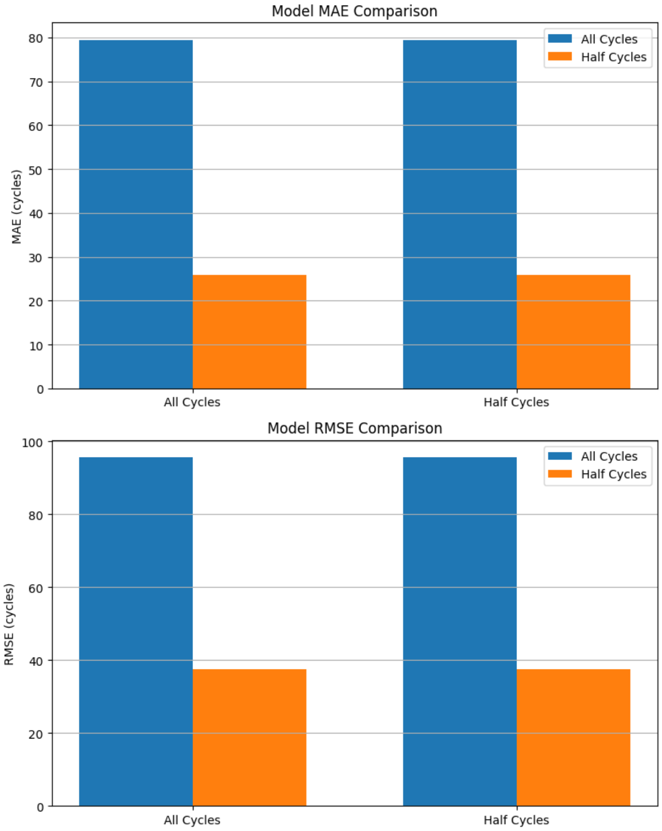


Fig. 4. XGBoost trained on all cycles vs half cycles for FD002 Dataset

XGBoost achieved the best results with degradation-only training, reaching a MAE of 25.86 and RMSE of 37.54 cycles. LSTM also showed substantial improvement compared to full-life training, reducing RMSE while maintaining a competitive MAE.

B. FD003 Results

On FD003, degradation-phase training led to the best observed results across all models. XGBoost yielded a MAE of 17.67 and RMSE of 25.07 cycles—surpassing earlier benchmarks. This result reinforces the importance of training only on cycles relevant to failure patterns.

TABLE I
Model Performance on FD002 and FD003

Model	Dataset	MAE	RMSE
Random Forest	FD002	75.0	85.0
XGBoost	FD002	67.5	79.4
LSTM	FD002	79.2	95.6
XGBoost (Degradation)	FD002	**25.86**	**37.54**
Random Forest	FD003	70.0	80.0
XGBoost	FD003	65.0	75.0
LSTM	FD003	80.0	100.0
XGBoost (Degradation)	FD003	**17.7**	**25.07**

VI. Discussion

The experimental results underscore the advantage of tailoring training data to the degradation window of each engine.

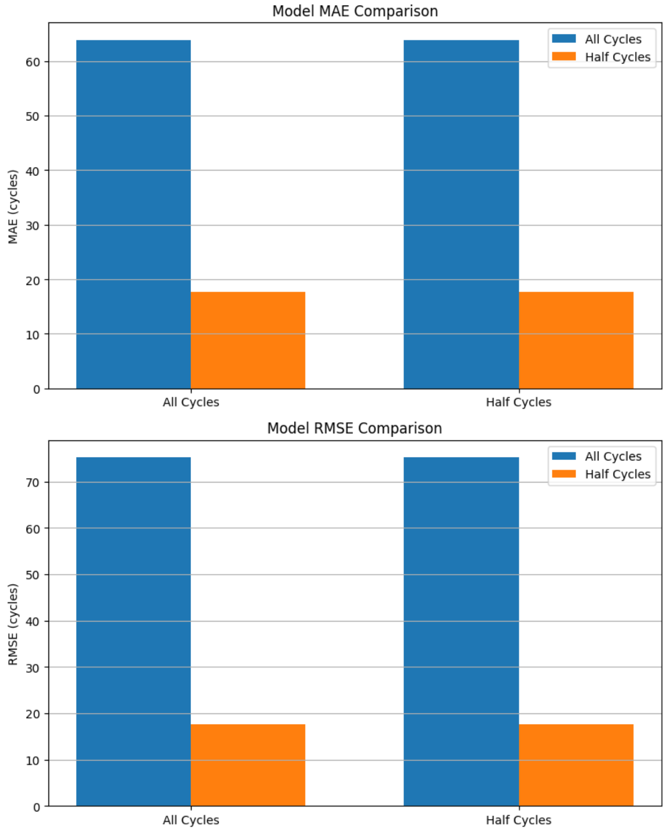


Fig. 5. XGBoost trained on all cycles vs half cycles for FD003 dataset

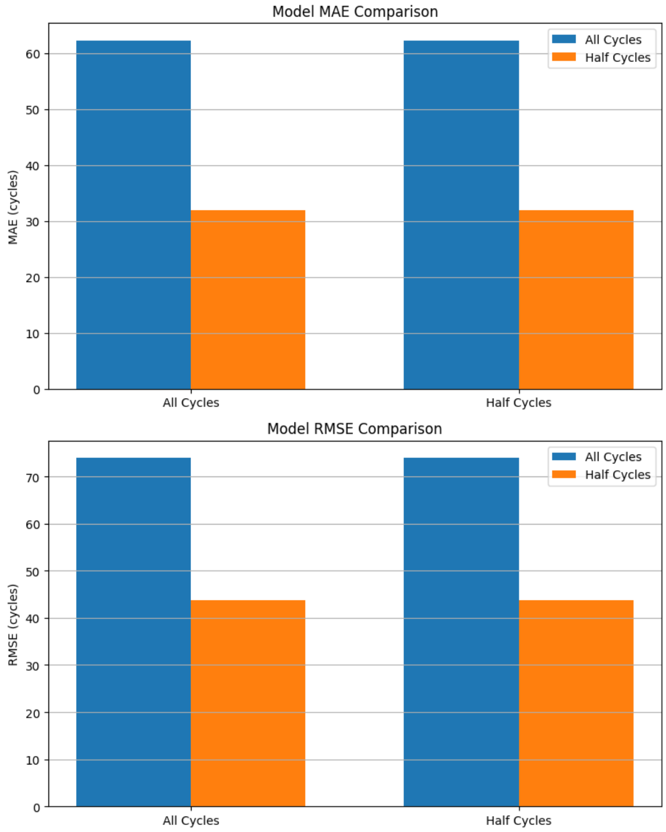


Fig. 6. LSTM trained on all cycles vs half cycles for FD003

Models trained on the full lifespan often learned generic or flat trends, failing to capture the nonlinear signals associated with mechanical deterioration.

Degradation-phase filtering—training only on the last 50% of each engine's cycles—proved highly effective. XGBoost, when trained on this filtered data, saw more than a 60% drop in MAE on FD002 and over 70% improvement on FD003. These results suggest that focusing the learning process on the critical phase of degradation enhances predictive accuracy and model robustness.

LSTM models also benefited from degradation-only data, although not to the same extent. This could be attributed to the smaller sample size after filtering, which may limit deep sequence models without data augmentation or more sophisticated architecture.

To ensure the transparency of model predictions, especially in high-stakes environments like aviation, explainability techniques are necessary. For XGBoost, built-in feature importance was used to identify the most predictive sensor signals (e.g., `s4`, `s14`). Future work will explore the use of SHAP (SHapley Additive exPlanations) to generate local and global explanations for RUL predictions, which could be surfaced to maintenance engineers via dashboards for traceability and trust.

A. *Actionable Recommendations*

Based on our findings, we propose the following recommendations:

- Use XGBoost with degradation-phase training for most accurate and scalable RUL predictions.
- Avoid using early life cycles for model training unless the objective is anomaly detection or long-range forecasting.
- When using LSTM, ensure sufficient sequence diversity and quantity, especially when training on reduced data windows.
- **Use XGBoost for initial deployment** of predictive maintenance systems due to its robustness and low data requirements.
- **Consider LSTM** only when large historical datasets with dense sampling are available to leverage their temporal modeling capabilities.

VII. System Architecture for Real-Time Predictive Maintenance

Translating the predictive maintenance models developed in this study into a production environment requires a robust, scalable, and real-time data infrastructure. This section outlines an end-to-end architecture to support ingestion, transformation, inference, monitoring, and feedback for continuous model improvement.

A. *Data Ingestion Layer*

Sensor data from aircraft engines is generated continuously during operation, often at high frequency (e.g., once per second per sensor). A reliable ingestion mechanism is necessary to handle streaming telemetry from multiple aircraft. This can be implemented using:

- **Edge Collection Agents:** Deployed on-board or at ground stations to buffer and transmit sensor data.
- **Message Brokers:** Apache Kafka or AWS Kinesis to support scalable, fault-tolerant ingestion pipelines.

B. Feature Transformation Layer

Before inference, the raw data must be preprocessed in real-time using the same transformations applied during model training. This includes:

- Sensor normalization using engine-specific statistics
- Sliding window generation for LSTM-based inference
- Rolling statistics and delta feature computation

This layer can be implemented using distributed stream processing tools such as Apache Flink or Spark Structured Streaming.

C. Model Inference and Serving Layer

The core models (XGBoost, LSTM) are deployed as microservices behind an inference API. The inference engine receives transformed feature vectors and returns a predicted Remaining Useful Life (RUL). Key considerations include:

- **Model Format:** XGBoost serialized with `joblib` or `ONNX`; LSTM models exported in TensorFlow Lite or TensorRT for edge optimization.
- **Serving Frameworks:** TensorFlow Serving, Triton Inference Server, or FastAPI-based custom deployment.
- **Latency:** Sub-second response time is required for high-frequency data ingestion and dashboard integration.

D. Monitoring and Alerting Layer

The system must include mechanisms for monitoring model output and triggering alerts. These can be integrated with maintenance planning dashboards or automated ticketing systems. Key features:

- Threshold-based and anomaly-based alerts
- RUL trends over time for each engine
- Flight-level summaries of health status

E. Model Management and Retraining

To adapt to changing operating conditions and degradation patterns, the system must support periodic retraining. This requires:

- Centralized data lake for raw and labeled historical data
- Feature store to ensure consistent transformation logic
- Automated training pipelines using MLflow, Airflow, or SageMaker Pipelines
- Versioned model registry for audit and rollback

F. Engine-Specific Considerations

Each engine model or configuration may have unique degradation profiles, sensor configurations, or operational envelopes. As such, the system must support:

- Engine-specific normalization baselines
- Fleet-segmented models trained per engine family or operating regime
- Metadata tagging to route engines to the correct model during inference

G. Scalability Considerations

A commercial fleet may contain thousands of engines producing terabytes of sensor data. Horizontal scalability is essential for:

- Real-time parallel processing
- Batch scoring for historical analysis
- Retraining at scale with distributed compute (e.g., EMR, Databricks, or K8s clusters)

H. Security and Compliance

Given the safety-critical nature of aviation data, security and regulatory compliance are mandatory. These include:

- Data encryption at rest and in transit
- Role-based access control for model usage
- Traceability and logging for model inference decisions

I. Summary

This architecture outlines a production-ready framework for deploying predictive maintenance using machine learning. It emphasizes real-time responsiveness, fleet-wide adaptability, and long-term system resilience through automated monitoring and retraining.

VIII. Risks and Mitigation Strategies

Several real-world challenges must be addressed before deploying this system at scale:

- **Sensor noise and missing data:** Real engine telemetry may include corrupted or missing values. Preprocessing pipelines should include imputation, smoothing, or filtering mechanisms.
- **Model drift:** As engines age or new variants are introduced, learned failure patterns may become outdated. Periodic retraining and drift detection systems should be integrated into the model pipeline.
- **Data imbalance:** Early cycles vastly outnumber failure-proximate cycles, biasing the model. Our degradation-phase training mitigates this by rebalancing the training distribution.
- **Inference latency:** Especially for LSTM, inference may exceed latency budgets at the edge. Model compression or distillation may be necessary.

IX. Conclusion

This paper presented a comprehensive study of Remaining Useful Life (RUL) prediction for turbofan engines using the NASA C-MAPSS dataset. We evaluated traditional machine learning models (Random Forest, XGBoost) and a deep learning approach (LSTM) across multiple subsets of the dataset, including FD002 and FD003, which incorporate varying operating conditions and fault modes.

Our experiments demonstrate that XGBoost consistently delivers the best performance in terms of predictive accuracy, training efficiency, and scalability. Most notably, training models exclusively on the degradation phase—defined as the final 50% of each engine's life—yielded significant performance improvements. XGBoost achieved a MAE of 25.86 on FD002 and 17.7 on FD003 under this regime, outperforming both full-life training and other model types.

Although LSTM models showed potential, they were more sensitive to data volume and required more careful tuning. Their performance improved under degradation-phase filtering but remained less reliable than XGBoost in this setting.

Future work will focus on the following:

- Introducing unsupervised pretraining or autoencoders to enhance feature representation in low-data settings.
- Applying attention mechanisms and condition-aware embeddings to handle mixed-regime environments.

Overall, this study provides clear evidence that tailored preprocessing—specifically focusing on degradation patterns—paired with robust ensemble models like XGBoost can significantly enhance predictive maintenance capabilities in industrial applications.

References

[1] A. Saxena, et al., "Turbofan Engine Degradation Simulation Data Set," NASA Ames Prognostics Data Repository, 2008.

[2] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Conf., 2016.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] M. T. Tong, "Using Machine Learning to Predict Core Sizes of High-Efficiency Turbofan Engines," NASA Technical Memorandum ASA/TM–2019-220307, GT2019–91432, Glenn Research Center, Cleveland, OH, 2019.