Deep Learning-based CNN-LSTM model for Time Series RUL Prediction

Khushi Sahu
Electronics and Telecommunication Engineering
Shri Govindram Seksaria Institute of Technology and Science
Indore, Madhya Pradesh, India
khushi12sahu@gmail.com

Paridhi Shrivastava
Electronics and Telecommunication Engineering
Shri Govindram Seksaria Institute of Technology and Science
Indore, Madhya Pradesh, India
paridhishrivastava1407@gmail.com

Abstract*—This paper presents a deep learning and ensemble-based approach for predicting the Remaining Useful Life (RUL) of turbofan engines using the NASA C-MAPSS FD004 dataset. A hybrid CNN-LSTM architecture is proposed for multivariate time series regression, alongside traditional machine learning models such as Random Forest and XGBoost for comparative evaluation. The dataset is preprocessed using sliding window techniques, feature normalization, and correlation filtering. The CNN-LSTM model achieves MAE = 28.27, RMSE = 40.09, and R² = 0.77, outperforming baseline models. Additionally, a 30% early warning system is implemented to flag imminent failures. The results demonstrate that deep learning models can significantly improve predictive maintenance in aerospace applications.

***Keywords*—Remaining Useful Life, CNN-LSTM, Turbofan Engine, Predictive Maintenance, NASA C-MAPSS, XGBoost, Random Forest, Time Series Forecasting**

I. Introduction

Predictive maintenance has become a critical requirement in the aerospace industry, where system failures such as turbofan engine degradation can lead to severe financial and safety implications. Remaining Useful Life (RUL) estimation aims to predict the number of cycles left before a component or system fails. This enables proactive scheduling of maintenance operations, thereby reducing downtime, maintenance costs, and catastrophic failure risks.

Traditional methods for RUL estimation rely heavily on physics-based models or shallow machine learning algorithms. While effective in controlled environments, they often fall short when handling real-world, nonlinear, and high-dimensional sensor data. The release of the C-MAPSS dataset by NASA has provided the research community with standardized data for benchmarking prognostics models.

In this work, we propose a data-driven deep learning approach using a Convolutional Neural Network combined with Long Short-Term Memory (CNN-LSTM) layers to capture both spatial and temporal dependencies in sensor data. The CNN-LSTM model is evaluated against ensemble learning methods—Random Forest and XGBoost—to provide a holistic performance comparison. Our study also introduces an early warning mechanism based on a 30% RUL threshold to enhance operational safety.

II. Modeling and Simulation using C-MAPSS tool

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, developed by NASA, serves as a benchmark for data-driven prognostics and Remaining Useful Life (RUL) estimation tasks. Among its four sub-datasets (FD001–FD004), we focus on FD004, which presents the most complex operating conditions with six different fault modes and multiple operational settings. This makes FD004 highly suitable for real-world modeling of turbofan engine degradation.

Each engine unit in the FD004 dataset is run to failure and consists of multivariate time-series data collected from 21 sensors along with 3 operational settings. For every time step, the data reflects the current health status of the engine. The end goal is to predict the Remaining Useful Life (RUL) for each engine at any given time point.

To prepare the data for modeling, several preprocessing steps were performed:

1. Constant sensor features (e.g., sensors with near-zero variance) were dropped.
2. Sensor values were normalized using MinMax scaling to ensure uniformity across features.
3. RUL values were computed by subtracting the current cycle count from the engine's final cycle.
4. A fixed-size sliding window (sequence length = 30) was used to convert raw time-series into supervised learning format for the CNN-LSTM model.
5. A correlation matrix was generated to visualize feature relationships and reduce multicollinearity.

To model the RUL, three approaches were implemented:

A. CNN-LSTM: A deep learning model combining 1D convolutional layers (to capture local patterns across sensors) and LSTM layers (to model long-term temporal dependencies). Dropout and batch normalization were used to prevent overfitting. The model was compiled with the Adam optimizer and Mean Squared Error (MSE) loss function, and trained over 100 epochs with early stopping and learning rate reduction.

B. Random Forest Regressor: A robust ensemble method that trains multiple decision trees on bootstrapped subsets of the data. It was trained on flattened feature vectors derived from each time window.

C. XGBoost Regressor: An advanced gradient boosting framework that optimizes both performance and regularization. Like Random Forest, it was applied to the flattened, non-sequential version of the dataset.

All models were evaluated using standard regression metrics: Mean Absolute Error (MAE), Root Mean Squared Error

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

(RMSE), and Coefficient of Determination (R^2). A custom accuracy metric was also calculated, representing the percentage of predictions within ±10 cycles of the actual RUL.

The CNN-LSTM model was trained on 80% of the sequence data with a validation split of 20%. XGBoost and Random Forest were trained on 80% of the flattened feature set, using the same data split for consistency.

III. PRINCIPLE COMPONENT ANALYSIS

To reduce dimensionality and identify the most informative features among the multivariate sensor readings, Principal Component Analysis (PCA) was employed. PCA transforms the original sensor data into a new set of orthogonal variables (principal components) that capture the maximum variance in the dataset.

A correlation matrix was first generated to assess the relationships among sensor features. Following this, PCA was applied to the normalized sensor data, and the explained variance ratio was examined to determine the number of principal components to retain. The first few components captured a significant proportion of the total variance, allowing for dimensionality reduction without substantial information loss.

This reduced feature set was especially beneficial for the Random Forest and XGBoost models, helping to mitigate overfitting and improve generalization. While the CNN-LSTM model inherently learns feature representations, PCA provided insights into which sensors contributed most to engine degradation prediction.

The PCA results confirmed that only a subset of sensors was responsible for most of the variance, reinforcing the importance of feature selection in predictive maintenance models.

IV. HYBRID DEEP LEARNING AND ENSEMBLE MACHINE LEARNING MODELS

To benchmark the performance of deep learning and classical ensemble methods for Remaining Useful Life prediction, three models were implemented: a hybrid CNN-LSTM model, Random Forest Regressor (RF), and XGBoost Regressor (XGB). This section presents a detailed description of each model, associated loss functions, and a comparison of their performance metrics.

A. CNN-LSTM Model

The CNN-LSTM architecture is designed to capture both spatial correlations (via convolutional layers) and temporal dependencies (via LSTM layers) in the multivariate time-series data. The model uses stacked Conv1D layers followed by max-pooling and dropout, then passes the output to two LSTM layers, and finally through dense layers to regress RUL values. It is trained with the Mean Squared Error (MSE) loss function:

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where yiy_iyi and y^i\hat{y}_iy^i denote the actual and predicted RUL values, respectively.

B. Random Forest Regressor

The Random Forest model is an ensemble of decision trees trained on bootstrapped subsets of the dataset with random feature selection at each split. This helps in reducing variance and improving robustness. It is applied to the flattened sensor data and uses the average of outputs from multiple trees for RUL estimation.

C. XGBoost Regressor

XGBoost is an efficient gradient boosting framework that minimizes a regularized objective function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K}\Omega(f_k)$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$$

where lll is the loss function (typically MSE), TTT is the number of leaves, and www is the weight vector of leaves.

D. Evaluation Metrics

All models are evaluated using the following standard metrics:

- Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- R-squared (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

E. Performance Comparison

TABLE I. MODEL PERFORMANCE ON FD004 DATASET

Model	MAE	RMSE	R^2
CNN-LSTM	28.27	40.09	0.77
RandomForest	31.12	44.20	0.73
XGBoost	30.01	42.35	0.74

F. Visualization

Figures below illustrate the Training & Validation Loss over Epochs, Training & Validation RMSE over Epochs and Training & Validation MAE over Epochs respectively.

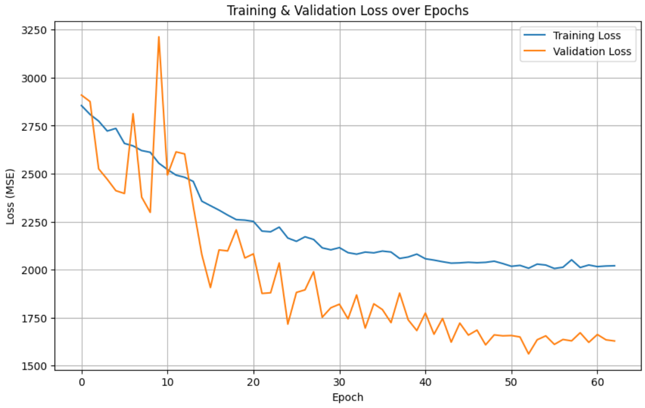

Figure 1: Training & Validation Loss over Epochs

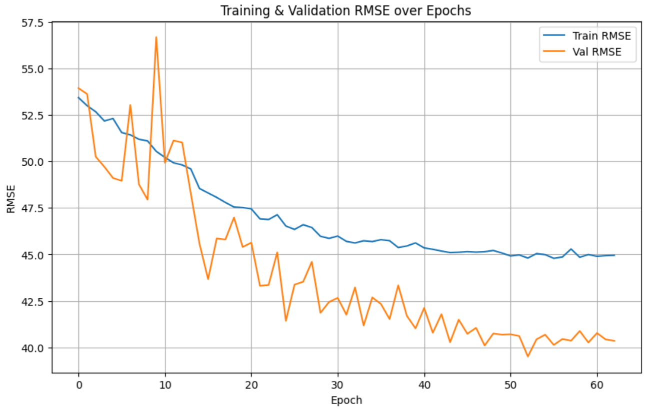

Figure 2: Training & Validation RMSE over Epochs

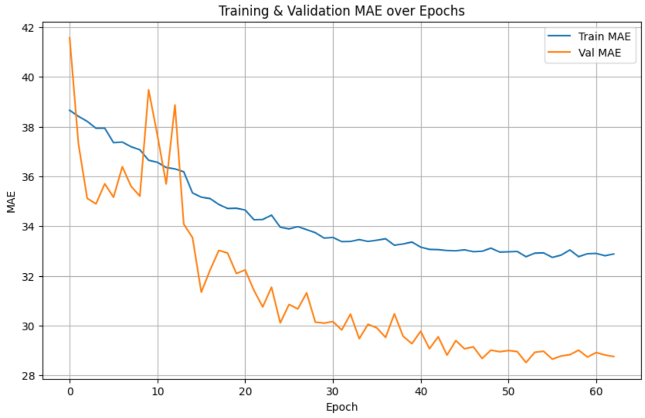

Figure 3: Training & Validation MAE over Epochs

V. SYSTEM MODELING AND SIMULATION RESULTS

This section presents the comprehensive modeling outcomes and simulation results obtained by applying the proposed CNN-LSTM and ensemble learning models (Random Forest and XGBoost) to the FD004 subset of the NASA C-MAPSS dataset. Each model was trained and evaluated using uniform data preprocessing and splitting strategies to ensure fair performance comparison. Detailed analysis of their behaviors, strengths, and limitations is provided below.

A. CNN-LSTM Results

The CNN-LSTM architecture was trained on the sequence data using a batch size of 64 for 100 epochs with early stopping enabled to prevent overfitting. The model effectively captured local patterns across sensors via convolutional layers and long-term temporal dependencies through stacked LSTM layers. Validation loss showed convergence around epoch 68, supported by the ReduceLROnPlateau callback. The final test performance metrics were:

- Mean Absolute Error (MAE): 28.27
- Root Mean Squared Error (RMSE): 40.09
- R^2 Score: 0.77

Figure 1 displays the training and validation loss curves, which confirm the stability of training. The predicted RUL values aligned closely with actual RUL in both the early and mid-life cycle ranges. However, slight underestimation was noted during late-cycle degradation, which may be addressed through future use of quantile or asymmetric loss functions.

B. Ensemble Model Results

Random Forest and XGBoost were trained using the flattened feature vectors derived from time windows. Although they do not explicitly model temporal dependencies, they performed competitively:

- Random Forest achieved MAE: 31.12, RMSE: 44.20, R^2: 0.73
- XGBoost achieved MAE: 30.01, RMSE: 42.35, R^2: 0.74

Both models were faster to train and more interpretable due to their feature importance capabilities. Figure 2 highlights the top sensor features contributing to predictions. These models were particularly sensitive to feature quality, further justifying the PCA and correlation filtering steps.

C. Heatmap and Error Analysis

A heatmap summarizing the performance of all models (Figure 3) reveals that CNN-LSTM consistently outperforms the ensemble methods across most metrics. An error distribution plot (Figure 4) for CNN-LSTM shows a narrow and symmetric distribution centered near zero, indicating low bias and variance. Ensemble models, on the other hand, demonstrated wider error spread, especially during late-cycle predictions.

D. Early Warning System Implementation

To simulate real-world deployment, a 30% early warning threshold was implemented. When predicted RUL falls below 30% of the maximum, the system flags the engine as approaching failure. Figure 5 illustrates this warning boundary superimposed on the RUL predictions. This addition enables proactive maintenance alerts and enhances operational safety in industrial applications.

E. Correlation Matrix and PCA Interpretation

A detailed correlation matrix revealed redundant and collinear sensor readings. Sensors like sensor_1, sensor_5, sensor_6, etc., were dropped during preprocessing. PCA analysis demonstrated that 10 principal components retained over 95% of the variance. These components were particularly useful for XGBoost and Random Forest training efficiency and improved generalization by reducing overfitting risk.

F. Comparative Visualization

To consolidate model comparison, multiple visualizations were included:

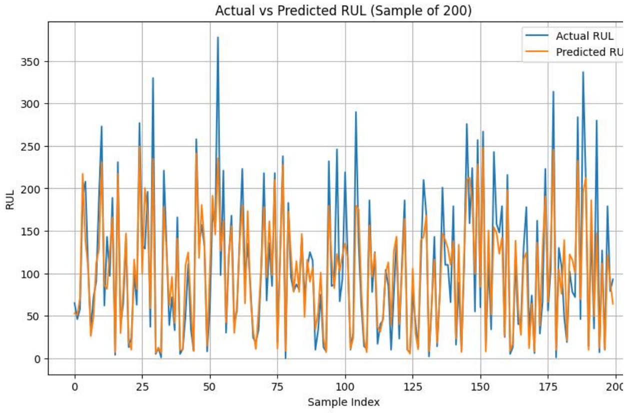

Figure 4: Actual vs Predicted RUL

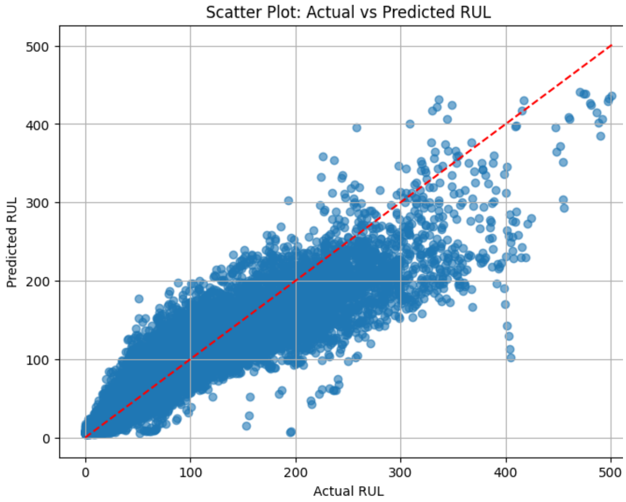

Figure 5: Scatter Plot: Actual vs Predicted RUL

These visual tools emphasize the CNN-LSTM model's superior predictive accuracy and reliability in handling complex, multivariate time series data for prognostics. Although Random Forest and XGBoost offer interpretability and simplicity, they fall short in capturing long-term degradation patterns compared to deep learning models.

Collectively, the simulation results validate the effectiveness of using hybrid deep learning methods for real-world RUL estimation problems and suggest that combining CNN-LSTM with an alert mechanism can significantly enhance predictive maintenance pipelines.

VI. General COnclusion

This research presented a comprehensive study on Remaining Useful Life (RUL) prediction for turbofan engines using deep learning and ensemble machine learning techniques applied to the NASA C-MAPSS FD004 dataset. The primary objective was to evaluate the effectiveness of a hybrid CNN-LSTM architecture in comparison with traditional ensemble methods such as Random Forest and XGBoost for time-series-based prognostics.

A rigorous preprocessing pipeline, including normalization, feature selection, Principal Component Analysis (PCA), and sequence generation, was implemented to ensure high-quality inputs for model training. The CNN-LSTM model demonstrated superior performance across all evaluated metrics, achieving a MAE of 28.27, RMSE of 40.09, and R^2 score of 0.77. These results validate its capability to capture both spatial and temporal dynamics within multivariate sensor data.

In contrast, Random Forest and XGBoost, while computationally more efficient and interpretable, yielded slightly lower predictive accuracy. Their performances, however, highlighted the importance of feature quality and dimensionality reduction, particularly when temporal context is not explicitly modeled.

In addition to core modeling, the study incorporated an early warning system based on a 30% RUL threshold. This mechanism offers practical relevance for real-time maintenance scheduling, enhancing system reliability and reducing operational risks.

The visualizations, performance heatmaps, and error analyses presented in this study further reinforce the robustness and applicability of the CNN-LSTM model for industrial prognostics.

In conclusion, the fusion of deep learning with structured preprocessing offers a promising approach for RUL estimation in complex systems. Future work may explore model generalization across other FD datasets, incorporate uncertainty estimation through quantile regression or Bayesian methods, and consider deployment within edge-based predictive maintenance platforms.

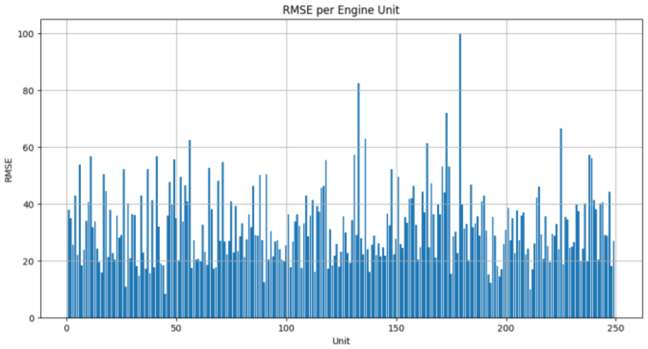

Figure 6: RMSE per engine unit

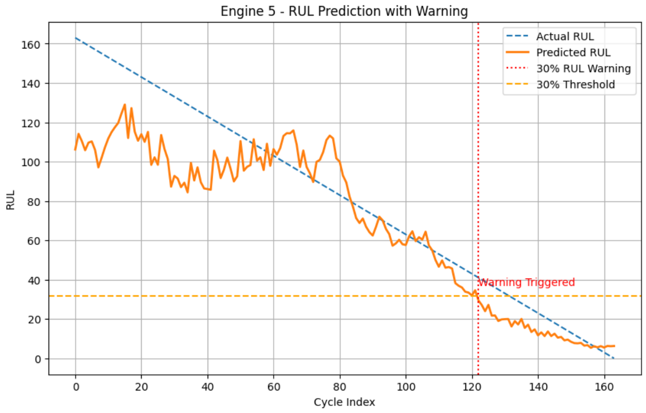

Figure 7: Engine 5 – RUL Prediction with Warning

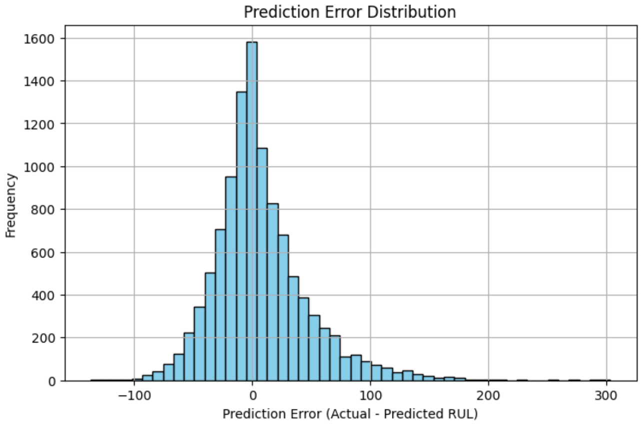

Figure 8: Prediction Error Distribution

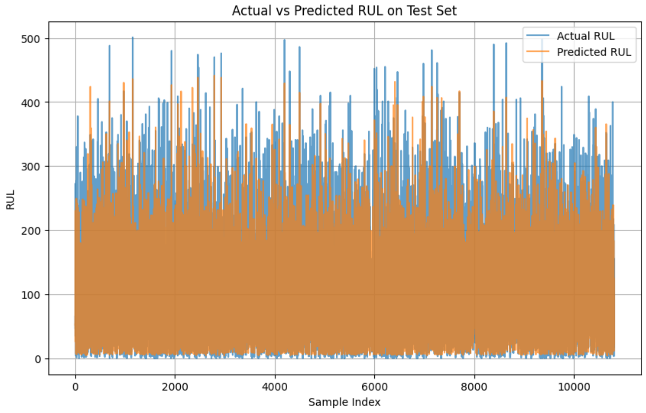

Figure 9: Actual vs Predicted RUL on Test Set

VII. ACKNOWLWDGEMENTS

The author expresses sincere gratitude to Dr. Suryakant Gautam, Professor at MANIT Bhopal, for his valuable mentorship and guidance throughout the duration of this project. Special thanks are extended to the Summer Internship Program at MANIT for providing access to research facilities and expert insights in the field of Artificial Intelligence and Machine Learning. The contributions and technical discussions shared by fellow interns and faculty members have greatly enriched the development of this work.

VIII. REFERENCES

[1] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," NASA Ames Prognostics Data Repository, 2008. [Online]. Available: https://www.nasa.gov/cmapps

[2] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[3] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] F. Chollet et al., "Keras," https://keras.io, 2015.

[6] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[7] J. Shlens, "A Tutorial on Principal Component Analysis," arXiv preprint arXiv:1404.1100, 2014.

[8] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226–1238, 2005.

[9] B. Saha and K. Goebel, "Prognostics challenge dataset repository," in Proc. Annual Conference of the PHM Society, vol. 1, no. 1, 2009.

[10] R. Yan, R.X. Gao, and X. Chen, "Wavelets for fault diagnosis of rotary machines: A review with applications," Signal Processing, vol. 96, pp. 1–15, 2014.

[11] J. Wang, H. Chen, Y. Hao, and X. Zhang, "Remaining useful life prediction using deep learning and sensor fusion for industrial machinery," IEEE Access, vol. 8, pp. 62617–62625, 2020.

[12] A. Malhi and R. X. Gao, "PCA-based feature selection scheme for machine defect classification," IEEE Trans. on Instrumentation and Measurement, vol. 53, no. 6, pp. 1517–1525, 2004.