Milestone 2 Report

1. Introduction

Milestone 2 focused on preparing the dataset for model training, engineering sequences suitable for deep learning architectures, implementing the RUL prediction model, and evaluating its performance. This stage transitioned the project from data exploration to actual predictive modeling. A significant amount of effort was spent on data normalization, sequence preparation, model architecture selection, and hyperparameter tuning to ensure accurate Remaining Useful Life (RUL) predictions.

---

2. Work Completed in Milestone 2

2.1 Combined Dataset Preparation

The four CMAPSS datasets (FD001–FD004) were consolidated into a single structured dataset to enable a unified model that generalizes across:

- Multiple operating conditions
- Multiple fault modes
- Different engine behaviors

Key steps:

- Merged FD001–FD004 using consistent column names
- Added dataset-specific one-hot encoded flags
- Ensured feature alignment across all datasets

This combined dataset allowed the model to learn broader degradation patterns instead of being limited to a single dataset.

2.2 Feature Scaling

To ensure all sensor readings contribute equally to model training, scaling was performed using:

- MinMaxScaler (or StandardScaler based on experiments)

Scaling was essential because:

- Sensor values vary significantly in magnitude
- Deep learning models require normalized input for stable training
- Prevents larger-valued sensors from dominating gradients

Separate scalers were created and saved for use during model inference.

2.3 Sequence Creation (Sliding Window Method)

Since turbofan sensor data is time-series, sequences were created using sliding windows.
Example:

Sequence length = 30 cycles
Input: 30 consecutive sensor readings
Output: RUL at the last cycle

This step was required because:

- LSTM/BiLSTM models need temporal context
- Predicting RUL depends on trend, not single points
- Sequence learning improves accuracy significantly

2.4 Model Architecture Development

A deep learning architecture suitable for RUL prediction was implemented. The chosen model was:

- Bidirectional LSTM (BiLSTM)
- Multiple stacked layers for deeper temporal learning
- Dense layers for final regression output
- EarlyStopping to prevent overfitting

Reason for choosing BiLSTM:

- Captures both past and future context
- Better representation of degradation patterns
- Performs better than plain LSTM or GRU in RUL prediction tasks

---

2.5 Model Training

The unified model was trained on combined FD001–FD004 data.
Important training parameters:

- Epochs: ~50–80
- Batch size: 64 or 128
- Sequence length: 30
- Loss function: MSE
- Optimizer: Adam

Training history showed:

- Steady decrease in training loss
- Validation loss stabilizing due to EarlyStopping
- No significant overfitting

2.6 Model Evaluation and Metrics

The trained model was evaluated using:

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- RUL prediction distribution
- Engine-wise RUL prediction curves

Observations:

- Validation loss and MAE remained within acceptable bounds
- Model performed consistently across different datasets
- Combined dataset training improved robustness and generalization

---

2.7 Visualization of Training Performance

Visualizations included:

- Training vs validation loss curves
- Sensor trends
- Cycle-wise degradation
- Heatmaps showing correlation across sensors

These provided insights into model behavior and data quality.

---

3. Summary of Milestone 2 Outcomes

Milestone 2 successfully delivered:

- Combined and normalized dataset ready for training
- Sliding-window sequences constructed for time-series learning
- BiLSTM-based model designed and trained
- Evaluation metrics confirming model accuracy
- Visualizations demonstrating sensor behavior and degradation

This milestone established a functioning RUL prediction model capable of generalizing across FD001–FD004.

---

4. Work Planned for Milestone 3

Milestone 3 will focus on deployment, explainability, and dashboarding. The objective is to make the system usable by engineers and stakeholders.

4.1 Model Deployment

- Save trained model in `.h5` format
- Implement inference pipeline for new engine data
- Ensure consistent scaling using saved scalers

4.2 Interactive Dashboard

A web-based predictive maintenance dashboard will be developed using:

- Flask (backend)
- HTML/CSS (frontend)
- Plotly for interactive visualizations

Dashboard features planned:

- File upload for engine data
- Prediction of RUL
- Sensor trend visualization
- Degradation curves
- Correlation heatmap
- Health status indicators
- Exportable PDF reports

4.3 Explainability of Predictions

Milestone 3 will include:

- Feature importance analysis
- SHAP or LIME explainability
- Interpretation of sensor contributions to RUL
- Visual comparison of predicted vs actual degradation

4.4 Performance Optimization

- Model re-tuning if needed
- Experimentation with hybrid architectures
- More robust sequence lengths
- Fine-tuning learning rate, dropout, activation functions

4.5 Validation with Realistic Scenarios

- Testing the model on unseen engine traces
- Evaluating misprediction regions
- Ensuring reliability for deployment

---

5. Conclusion

Milestone 2 focused on building a stable and accurate RUL prediction model through dataset integration, feature scaling, sequence preparation, and LSTM-based modeling. The results indicate strong model performance with controlled training behavior and meaningful insights. Milestone 3 will expand on this by building a deployment-ready dashboard and adding model explainability and reporting capabilities.