Milestone 2 — Model Development & Training

Project: PrognosAI — AI-Driven Predictive Maintenance System
By: Nithin G J
Date: October 24, 2025

■ Objective

To build and train a deep learning model that predicts the Remaining Useful Life (RUL) of industrial machinery using multi-sensor time-series data from the NASA CMAPSS dataset. This model helps forecast machine failures in advance, plan maintenance proactively, and reduce downtime.

■■ Implementation Summary

A time-series model was developed using LSTM (Long Short-Term Memory) networks. The approach includes loading the CMAPSS dataset, computing RUL values, filtering unnecessary sensors, scaling features, adding trend-based features, creating sequences of 30 time steps for LSTM input, and training with 5-fold cross-validation.

■ Model Architecture

The model uses Bidirectional and stacked LSTM layers for sequence learning.
1. Input layer (sequence length = 30)
2. BiLSTM (128 units) → Dropout (0.3) → BatchNorm
3. BiLSTM (64 units) → Dropout (0.3) → BatchNorm
4. LSTM (32 units) → Dropout (0.2)
5. Dense (64, ReLU) → Dropout (0.2)
6. Dense (1) — Output RUL prediction
Optimizer: Adam (lr=0.001), Loss: MSE, Metric: MAE

■ Key Parameters

Parameter	Value
FD_NUMBER	1
Window Size	30
Cross-Validation Folds	5
Epochs	50
Batch Size	64
Dropout	0.3 / 0.3 / 0.2 / 0.2
Scaling	MinMaxScaler
Trend Features	mean, std, diff, rolling_mean(5)

■ Data and Preprocessing

Files Used: train_FD001.txt, test_FD001.txt, RUL_FD001.txt. Steps include computing RUL per engine, adjusting test RULs, dropping irrelevant sensors, generating trend features, and creating time-windowed sequences for LSTM input.

■ Training & Evaluation

The model was trained using 5-fold K-Fold cross-validation. EarlyStopping and ReduceLROnPlateau callbacks were used. The final model was trained on the full dataset and evaluated on the test set using RMSE and MAE metrics.

■ Outputs

- Trained model: models_m2/optimized_fd1.h5
- Loss curve: graphs_m2/loss_curve_fd1.png

■ How to Run

Install dependencies and run the training script:

```
pip install numpy pandas scikit-learn matplotlib tensorflow keras
```

1. Set BASE_PATH to your dataset folder
2. Run the training file (milestone_2.py)
3. Check outputs in models_m2 and graphs_m2 folders

■ Observations & Next Steps

Use GroupKFold to prevent data leakage, save model history and scaler, test GRU/1D-CNN alternatives, perform hyperparameter tuning, and deploy the model as a Streamlit or TensorFlow web app.

■■■ Author

Nithin G J – Project Developer, PrognosAI