Predictive Maintenance of Industrial Machinery Using Machine Learning

Challenges

- Unexpected component failures under varying operational conditions lead to unplanned downtime, financial losses, and safety risks.
- Diverse operating conditions increase the complexity of predicting Remaining Useful Life (RUL), requiring tailored approaches for accuracy.
- Developing a unified model that generalizes effectively across all datasets while maintaining predictive accuracy is a significant challenge.

Goals

- Build dataset-specific models leveraging machine learning to capture unique operational behaviors and improve predictive accuracy.
- Develop a unified model capable of generalizing RUL predictions across diverse operating conditions.
- Address operational efficiency and safety concerns by ensuring timely maintenance and minimizing unplanned downtime.

Aim

Aim-1:

- Develop **three distinct machine learning models** (Random Forest, XGBoost, and Ensemble model) tailored for each dataset (FD001, FD002, FD003, FD004), addressing their unique operational conditions.
- Perform a comparative analysis of the model performances to determine the most effective model for predicting the Remaining Useful Life (RUL) under specific operating environments by categorizing predictions into **Critical, Warning**, or **Healthy**.

Aim-2:

- Develop a **unified machine learning model** capable of predicting the Remaining Useful Life (RUL) across all datasets (FD001, FD002, FD003, FD004) under diverse operational conditions.
- Ensure the unified model is flexible, robust, and adaptable to varying scenarios.
- Compare the performance of the unified model against the best-performing individual models to evaluate its effectiveness and generalization capabilities across different datasets.
- Highlight the trade-offs between dataset-specific accuracy and the versatility of the unified model.

Dataset

- NASA's Turbofan Engine Degradation Simulation Dataset
- **C-MAPSS** stands for **Commercial Modular Aero-Propulsion System Simulation**.
- It is a high-fidelity simulation environment developed by NASA to simulate the performance and behavior of commercial aircraft engines under various conditions.
- C-MAPSS is widely used for predictive maintenance research, particularly for modeling engine degradation and forecasting remaining useful life (RUL).

- Total rows (train + test): **265,256**
- The dataset contains 26 rows where:
  - **1st column**: Unit number (engine ID)
  - **2nd column:** Cycle number. This tracks how many operational cycles the engine has gone through (e.g., 1, 2, 3, ...).
  - **3rd to 5th columns**: Operational settings.
  - **6th to 26th columns**: Sensor measurements 1 through 26. These are the sensor readings captured for each engine during each cycle.

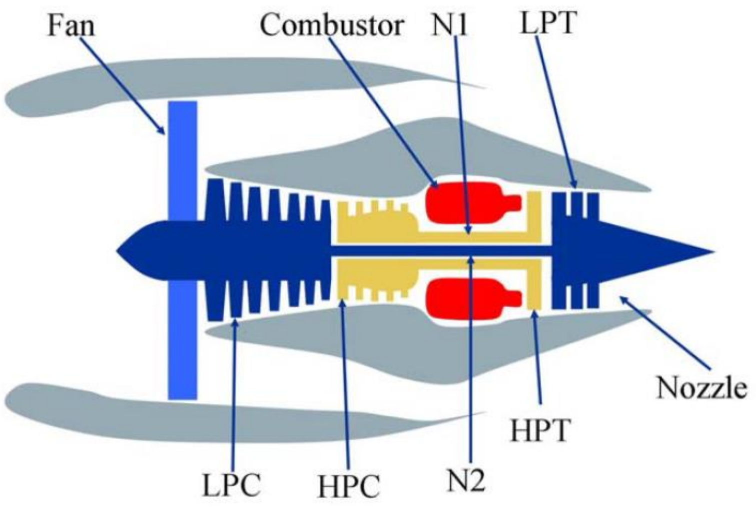
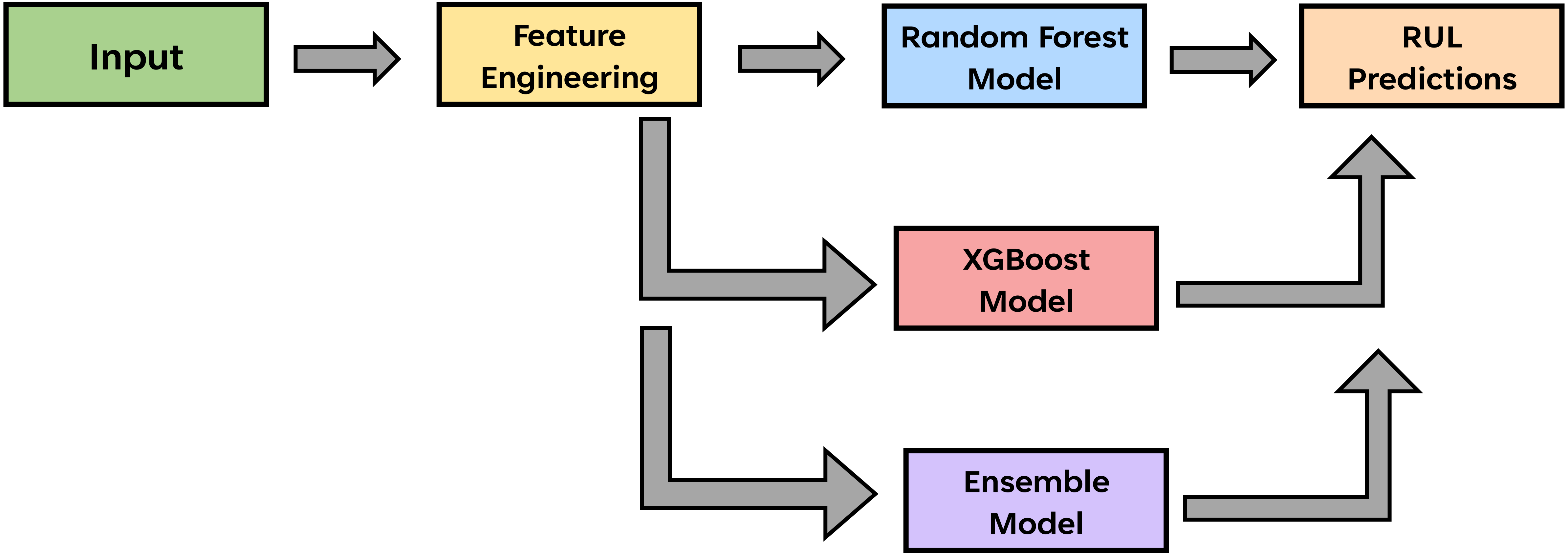


Fig1: Simplified diagram of engine simulated in C-MAPSS (Commercial Modular Aero-Propulsion System Simulation)

Aim 1: Building and Comparing Individual Models for Each Dataset

Developing and comparing three machine learning models (Random Forest, XGBoost, and Ensemble model) for each dataset (FD001, FD002, FD003, FD004) tailored to their specific operational conditions

Workflow of Machine Learning Models for RUL Prediction



Input

The **input** consists of sensor and operational data collected from engines under various operating conditions.

Feature Engineering

- Performed correlation analysis to identify redundant and insignificant features using the correlation matrix.
- Identified important sensors based on higher correlation with RUL and operational cycles.
- Dropped redundant features with low correlation or overlap.
- Engineered rolling statistics for retained sensors using Rolling mean, rolling standard deviation, and rolling difference were computed to capture temporal trends.

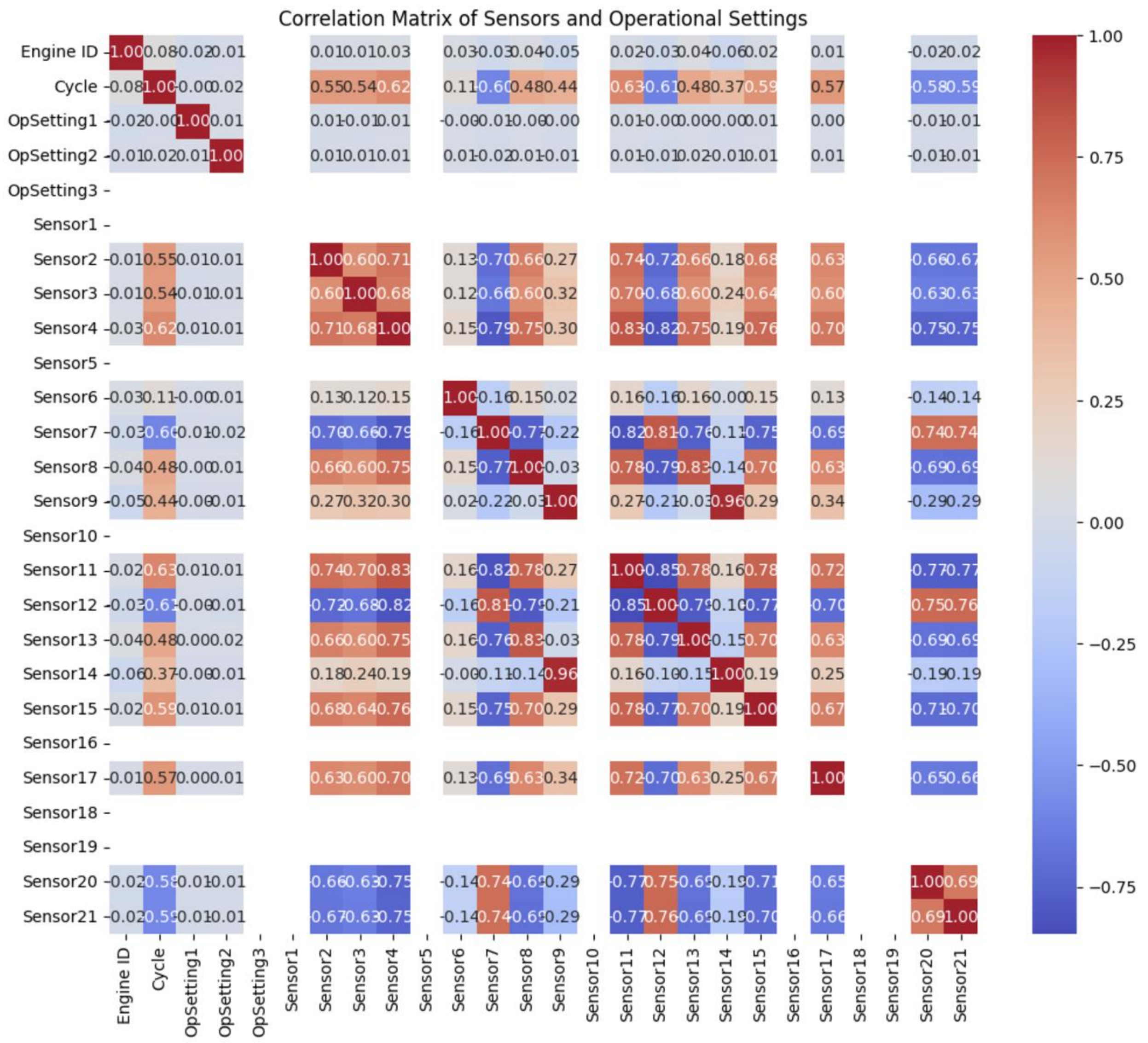


Fig2. Correlation Matrix of Sensors and Operational Settings

Random Forest Model Architecture

Ensemble of Decision Trees: Comprises multiple independent decision trees trained on bootstrapped subsets of the data. Each tree captures patterns in a randomly sampled portion of the dataset.

Training:

- Preprocessed X_train (sensor and operational data) and y_train (Remaining Useful Life, RUL).
- Training involves building 200 decision trees, each trained on bootstrapped subsets of the data.

Key Parameters:

- Number of Trees (n_estimators): 200
- Maximum Depth (max_depth): 10
- Random State (reproducibility): 42

Prediction:

- Each tree predicts RUL for test samples.
- Final RUL prediction is the average of all tree predictions.

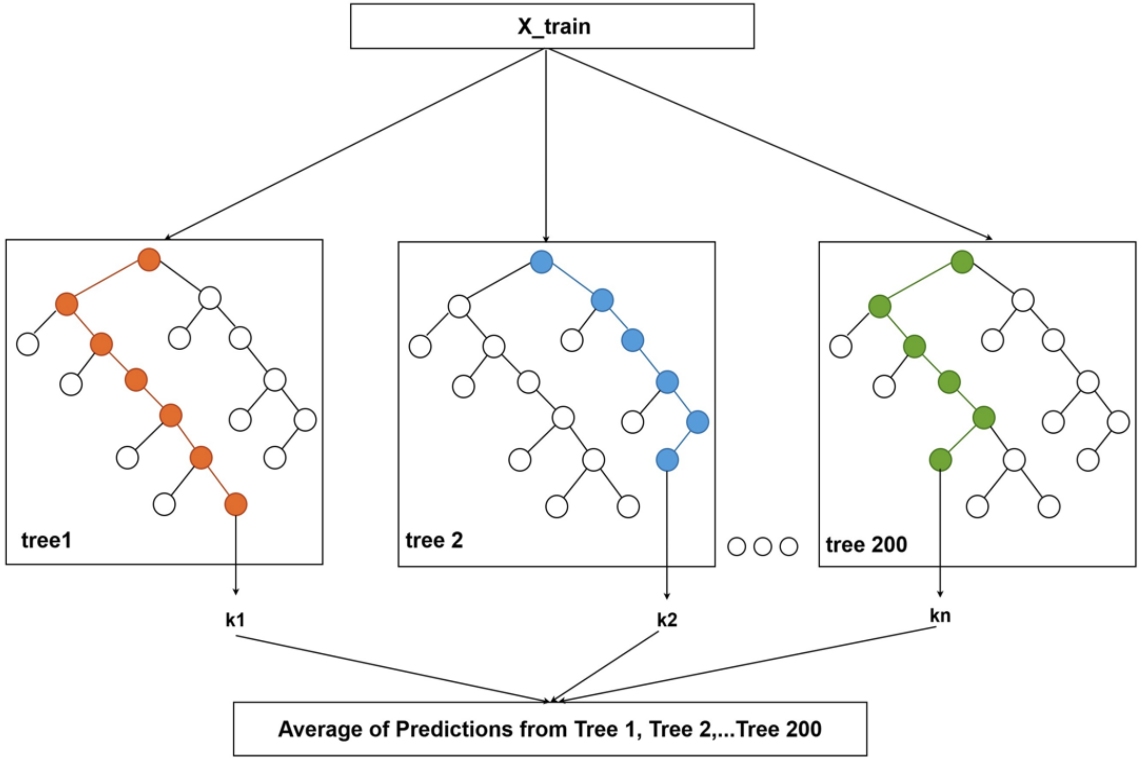


Fig3. Random Forest Model Architecture

XGBoost Model Architecture

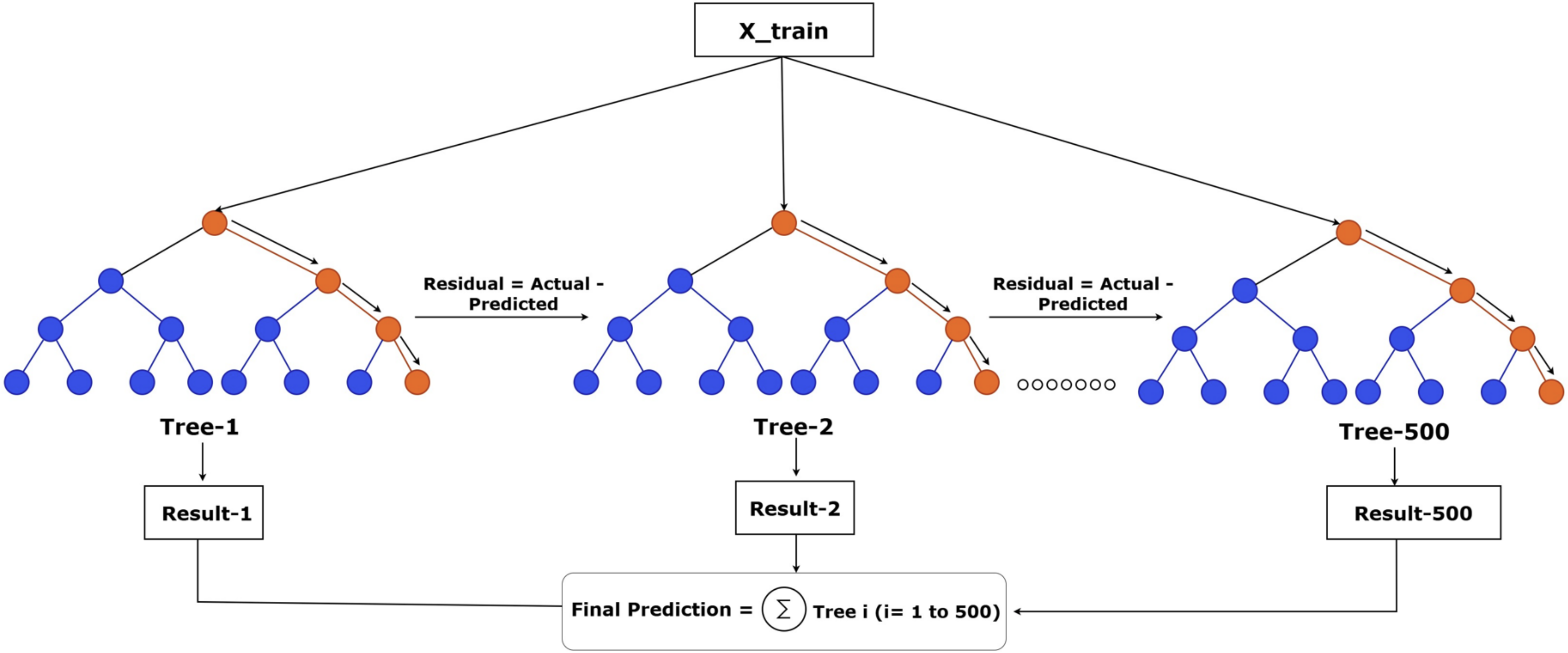


Fig4. XGBoost Model Architecture

XGBoost Model Architecture

- XGBoost is a gradient-boosted tree ensemble model that builds decision trees sequentially to minimize the residual error of the previous trees.
- Each tree in the ensemble focuses on correcting the errors (residuals) made by the previous trees.
- Trees are trained sequentially, with each tree focusing on reducing the residuals (difference between actual RUL and predicted RUL).
- The final prediction is the sum of weighted contributions from all trees.

Key Parameters:

- Number of Trees (n_estimators): 500
- Tree Depth (max_depth): 10
- Learning Rate (learning_rate): 0.05
- Subsampling (subsample): 0.8 (80%)
- Feature Sampling (colsample_bytree): 0.8 (80%)
- Random State: 42

Blended Ensemble Model Architecture

Model Overview
The blended ensemble model combines predictions from three base models:
1.Random Forest Regressor (RF)
2.XGBoost Regressor (XGB)
3.LightGBM Regressor (LGB)

These models are trained independently on the same training data but differ in their internal architecture and methodology. The ensemble model uses **optimized weights** to blend their predictions, resulting in a final prediction that leverages the strengths of all three models.

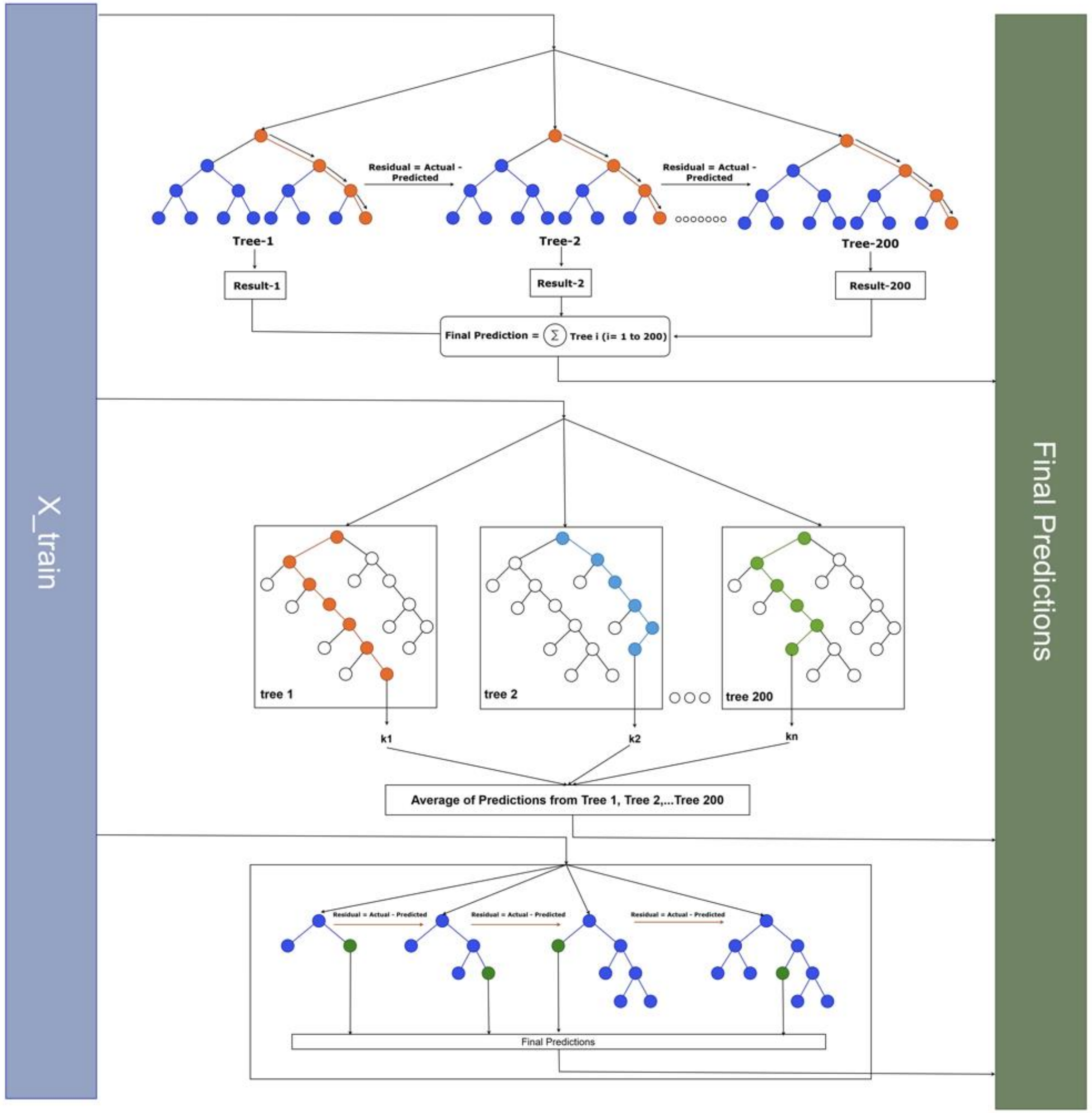


Fig5. Blended Ensemble Model Architecture

Input Data Processing: Input data includes sensor measurements and operational settings for engines. Data preprocessing includes scaling features using StandardScaler and feature engineering with rolling statistics (mean, standard deviation, and differences).

Model Training

Random Forest Model:

- 200 trees (n_estimators=200) with a maximum depth of 10.
- Bootstrapped sampling of data for each tree.

XGBoost Model:

- 200 trees (n_estimators=200) with a maximum depth of 5.
- Gradient boosting framework with a learning rate of 0.1.

LightGBM Model:

- 200 trees (n_estimators=200) with a maximum depth of 7.
- Optimized histogram-based learning and feature sampling.

Prediction Generation

Each model independently predicts Remaining Useful Life (RUL) for the test data.

Output predictions:

- RF_Predictions
- XGB_Predictions
- LGB_Predictions

Blending Mechanism

•A weight optimization process (using scipy.optimize.minimize function) is performed to determine the weights w1, w2, w3 for each model.

•Blended Prediction Formula: Blended Prediction is calculated using weighted contributions from each model:

$$\textbf{Blended Prediction} = w_1 * \textbf{RF Prediction} + w_2 * \textbf{XGB Prediction} + w_3 * \textbf{LGB Prediction}$$

Where:

- w_1, w_2, w_3: Weights for each model.
- RF Prediction, XGB Prediction, LGB Prediction: Predictions from Random Forest, XGBoost, and LightGBM, respectively.

Constraints:

- $w_1 + w_2 + w_3 = 1$ (weights must sum to 1).
- $0 \leq w_1, w_2, w_3 \leq 1$

FD001 Dataset Results						
Model	**Healthy (Predicted)**	**Warning (Predicted)**	**Critical (Predicted)**	**Healthy (Actual)**	**Warning (Actual)**	**Critical (Actual)**
Random Forest	72	24	4	67	28	5
XGBoost	72	25	3	67	28	5
Blended Model	73	23	4	67	28	5

FD002 Dataset Results						
Model	**Healthy (Predicted)**	**Warning (Predicted)**	**Critical (Predicted)**	**Healthy (Actual)**	**Warning (Actual)**	**Critical (Actual)**
Random Forest	69	27	3	66	27	7
XGBoost	68	25	8	66	27	7
Blended Model	68	28	4	66	27	7

FD003 Dataset Results						
Model	**Healthy (Predicted)**	**Warning (Predicted)**	**Critical (Predicted)**	**Healthy (Actual)**	**Warning (Actual)**	**Critical (Actual)**
Random Forest	75	21	4	71	24	5
XGBoost	82	18	-	71	24	5
Blended Model	74	23	3	71	24	5

FD004 Dataset Results						
Model	**Healthy (Predicted)**	**Warning (Predicted)**	**Critical (Predicted)**	**Healthy (Actual)**	**Warning (Actual)**	**Critical (Actual)**
Random Forest	72	28	0	68	28	4
XGBoost	69	25	5	68	28	4
Blended Model	71	27	2	68	28	4

Aim 2: Building and Comparing Unified Machine Learning Model

Develop a ***unified machine learning model*** *capable of predicting the Remaining Useful Life (RUL) across all four datasets (FD001, FD002, FD003, FD004).*

Model Overview

The blended ensemble model combines predictions from three powerful machine learning models—**XGBoost**, **Random Forest**, and **LightGBM**—to predict the Remaining Useful Life (RUL) of engines. The model leverages the strengths of each base model:

- **XGBoost** captures complex patterns through gradient boosting.
- **Random Forest** ensures stability and reduces overfitting through averaging.
- **LightGBM** handles large datasets efficiently with fast computation.

Model Architecture

1. **Input Data**:
    1. Features derived from engine sensor readings and operational settings.
    2. Rolling statistics (mean, standard deviation, and differences) added to enhance the model's understanding of trends over time.
2. **Base Models**:
    1. **XGBoost**:
        1. Parameters: 200 estimators, max depth of 7, learning rate of 0.1.
        2. Captures non-linear interactions in the data.
    2. **Random Forest**:
        1. Parameters: 200 estimators, max depth of 10.
        2. Provides stable predictions and reduces variance.

3. LightGBM:

1. Parameters: 200 estimators, max depth of 7, learning rate of 0.1.
2. Optimized for computational efficiency and speed.

Blending Mechanism

•A weight optimization process (using scipy.optimize.minimize function) is performed to determine the weights w1, w2, w3 for each model.

•Blended Prediction Formula: Blended Prediction is calculated using weighted contributions from each model:

$$\textbf{Blended Prediction} = w_1 * \textbf{RF Prediction} + w_2 * \textbf{XGB Prediction} + w_3 * \textbf{LGB Prediction}$$

Where:

• w_1, w_2, w_3: Weights for each model.

• RF Prediction, XGB Prediction, LGB Prediction: Predictions from Random Forest, XGBoost, and LightGBM, respectively.

Constraints:

• $w_1 + w_2 + w_3 = 1$ (weights must sum to 1).

• $0 \leq w_1, w_2, w_3 \leq 1$

Results

Model	Healthy (Predicted)	Warning (Predicted)	Critical (Predicted)	Healthy (Actual)	Warning (Actual)	Critical (Actual)
Unified Model	71%	27%	2%	67%	27%	4%

The diverse operational conditions across datasets make it challenging for a unified model to generalize effectively and face difficulty due to varying failure patterns and operational distributions across datasets.

Conclusion

- Individual models outperform unified models by capturing the unique operational conditions and failure behaviors specific to each dataset.
- Unified models dilute predictive accuracy when applied to datasets with distinct trends and varying characteristics.
- Dataset-specific tailored models excel in handling diversity, making them more effective for datasets with unique conditions.
- Unified models are more suitable for datasets with shared or similar operational environments.