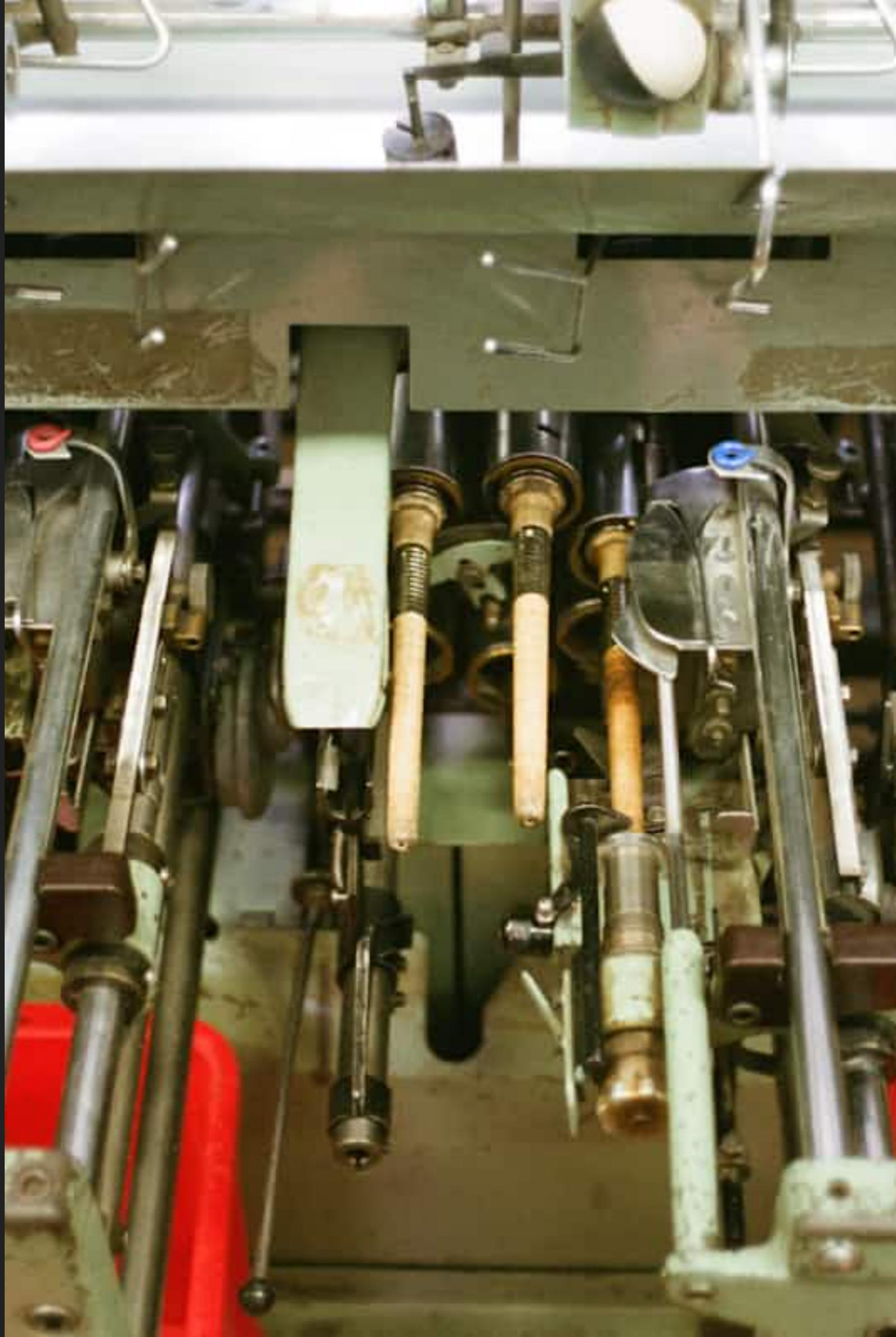
Predictive Maintenance for Industrial Equipment

Revolutionizing industrial operations by transforming reactive maintenance into proactive, data-driven strategies. This project tackles the critical challenge of unplanned equipment downtime, a major source of inefficiency and financial loss in heavy industries.



The Business Imperative: Mitigating Unplanned Downtime

Increased Repair Costs

Unexpected machine failures lead to costly emergency repairs and component replacements, often at premium prices due to urgency.

Production Disruptions

Breakdowns halt production lines, causing delays, missed deadlines, and potential loss of customer trust and revenue.

Logistical Challenges

Disruptions extend to supply chains and logistics, impacting delivery schedules and overall operational flow.

Unplanned equipment downtime is a primary driver of operational inefficiency and significant financial loss in industries reliant on heavy machinery. Each unexpected failure introduces cascading negative impacts, from costly emergency repairs to severe disruptions in production schedules and logistics. This project was initiated as a direct, data-driven response to this critical business challenge, aiming to shift the maintenance paradigm from reactive repair to proactive, predictive intervention.

Problem Statement: Shifting from Reactive to Predictive

The core business challenge was articulated by the operations and engineering division, which identified the dual financial and operational impact of unexpected machine failures. These events simultaneously "increase repair costs and cause disruptions in production and logistics." The absence of a reliable forecasting mechanism meant that maintenance was often performed only after a breakdown, incurring maximum cost and operational disruption.

This project aims to address the critical need for a predictive maintenance tool. By forecasting equipment health, we can enable maintenance teams to take proactive action, preventing costly failures and ensuring smoother operations. The goal is to move beyond traditional reactive maintenance, where repairs occur only after a breakdown, to a system that anticipates issues before they arise.

Project Objective: Empowering Proactive Maintenance

The ML Team was tasked with developing a predictive maintenance tool to address this problem. The central goal was to create a machine learning model capable of forecasting equipment health and providing actionable warnings.

Estimate Health Score or RUL

Develop a model to estimate a machine's health score or its Remaining Useful Life (RUL).

Flag Impending Failure Risk

Clearly identify and flag any impending failure risks to allow for timely intervention.

Deployable Lightweight Frontend

Ensure the system is deployable through a lightweight frontend for accessible use by engineers and operations staff.

This report details the technical methodology employed to build a solution that meets these objectives, from data preparation to the dual-strategy modeling approach.

Technical Methodology: Data-Driven Insights

A successful predictive model is not merely an algorithm; it is a comprehensive system built on robust data and meticulous analysis. Our approach involved leveraging historical sensor and operational time-series data to develop a sophisticated machine learning model.

1. **Data Acquisition**

   Gathering relevant historical sensor and operational time-series data.

2. **Data Preprocessing**

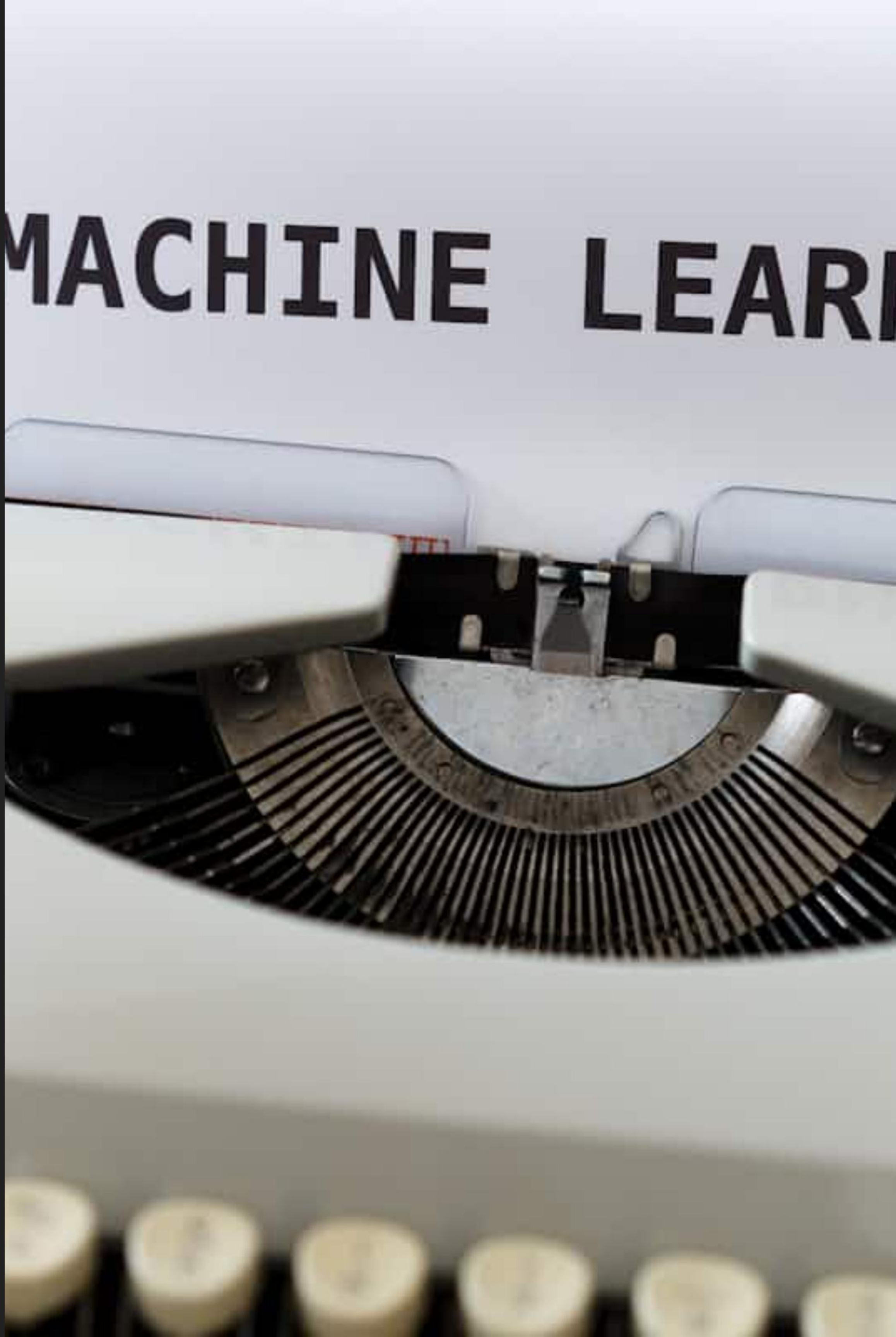
   Cleaning, transforming, and preparing data for model training.

3. **Model Development**

   Building and training machine learning and deep learning models.

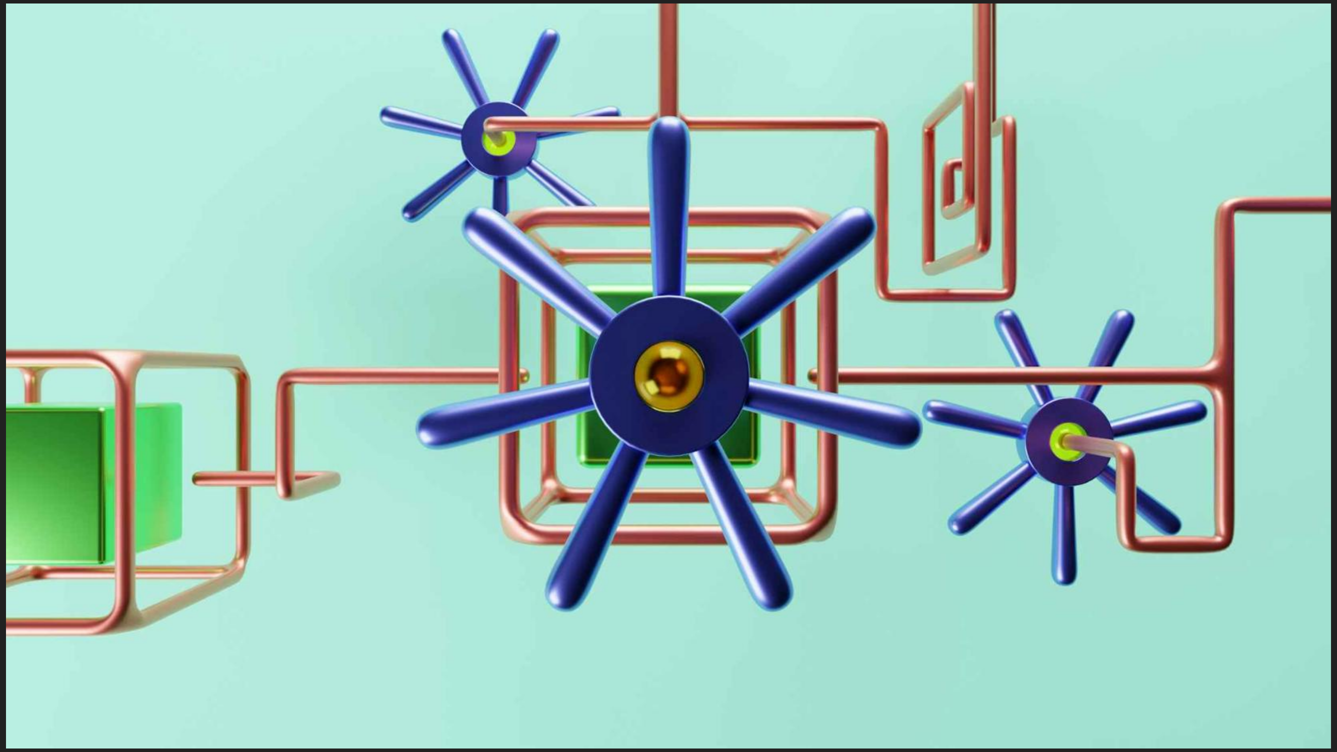
4. **Evaluation & Deployment**

   Assessing model performance and integrating into a user-friendly interface.



Data Foundation: NASA C-MAPSS Dataset

Our predictive maintenance solution is built upon the robust NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) FD001 dataset, a widely recognized benchmark for prognostics and health management. This dataset provides a rich source of time-series data from turbofan engines, simulating various operational conditions and degradation patterns.



- **Dataset: NASA C-MAPSS (FD001 subset)**

  A standard dataset for Remaining Useful Life (RUL) prediction.

- **Expected Files**

  `train_FD001.txt`, `test_FD001.txt`, `RUL_FD001.txt` (within a `CMaps` folder).

- **Key Columns**

  `engine_id`, `time_in_cycles`, `op_setting_1..3`, `sensor_1..sensor_21`.

The dataset's detailed sensor readings and operational settings allow for a comprehensive analysis of engine degradation, crucial for accurate RUL estimation.

Data Preparation & Feature Engineering

The initial phase of the project focused on meticulous data preparation and insightful feature engineering, laying the groundwork for effective model training.

1 Data Loading & Validation

Loaded training, test, and RUL files from the C-MAPSS FD001 dataset, defining column names and verifying file paths.

2 Column Filtering

Dropped constant-value columns (those with a single unique value) to reduce noise and improve model efficiency.

3 RUL Calculation & Capping

Calculated training RUL as `max_cycle - time_in_cycles` and capped RUL at 125 cycles to manage extreme values.

4 Reproducibility Seeds

Set reproducibility seeds (`PYTHONHASHSEED`, `random`, `numpy`, `tensorflow`) with `SEED_VALUE = 42` for consistent results.

These steps ensure that the data fed into our models is clean, relevant, and optimized for predictive accuracy.

EDA & Feature Building: Crafting Model Inputs

Exploratory Data Analysis (EDA) and feature building were crucial for understanding the dataset's nuances and preparing it for advanced modeling.

Dataset Inspection

Basic dataset inspection, including shapes, per-engine cycles, and feature distributions, to gain initial insights.

Feature Selection

Selected relevant feature columns by excluding `engine_id`, `time_in_cycles`, and `RUL`.

Sequence Generation

Implemented `generate_sequences()` to produce sliding-window sequences for the LSTM model, with a chosen `sequence_length = 50`.

Feature Scaling

Scaled features using `MinMaxScaler` (fitted on training features and applied to test features) to normalize data ranges.

Test Set Preparation

For the test set, extracted the last `sequence_length` measurements per engine; shorter sequences were zero-padded at the beginning.

Model Training: Dual-Strategy Approach

We implemented a mix of machine learning and deep learning models to tackle the predictive maintenance problem, employing a dual-strategy approach for both early warning and RUL prediction.

Classification Models: Early Warning System

Random Forest

Robust ensemble method for identifying complex failure patterns.

Gradient Boosting

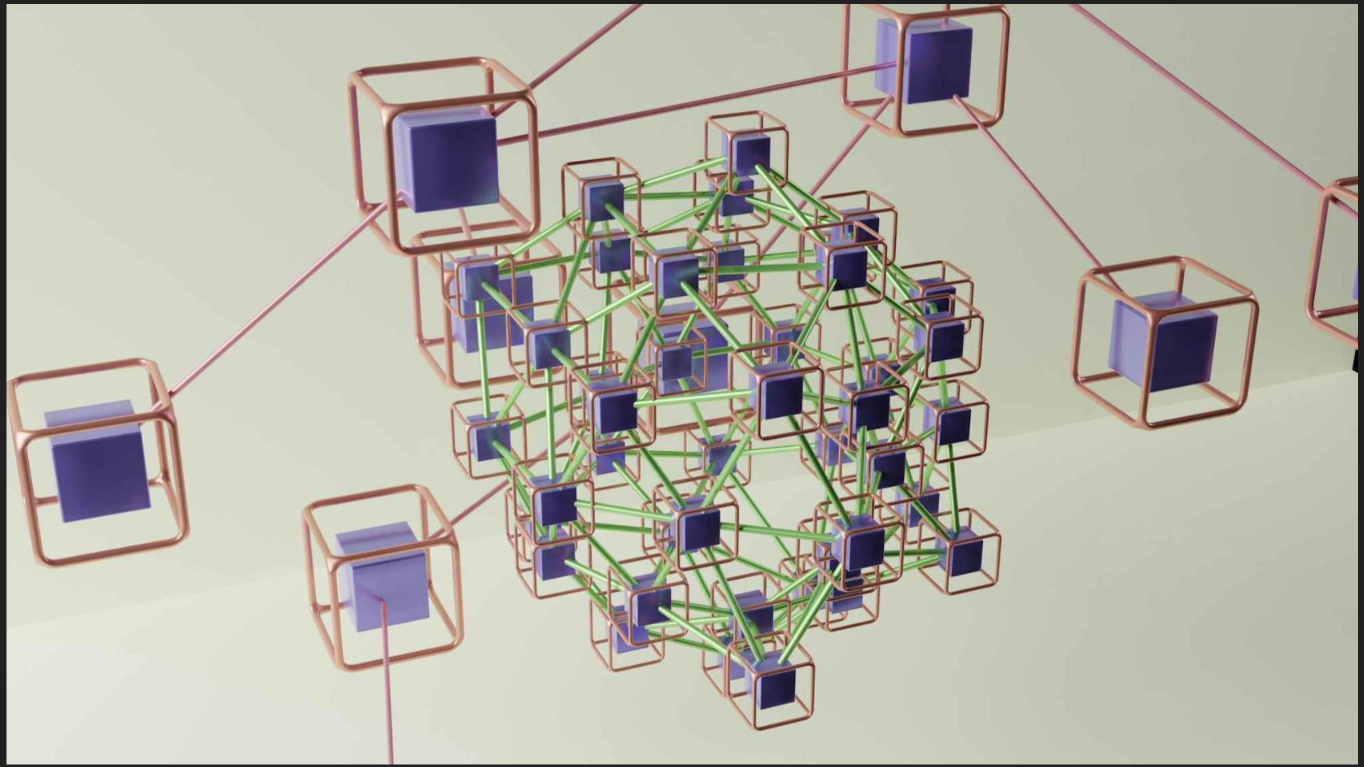
Powerful sequential model for improved accuracy in classification.

Support Vector Machine (SVM)

Effective for high-dimensional data, providing clear separation of failure classes.

These models were designed to provide an early warning of potential failures, each offering different strengths in handling complex data.

Deep Learning Model: RUL Prediction



An LSTM (Long Short-Term Memory) network was built to predict the Remaining Useful Life (RUL) as a continuous value. The architecture included stacked LSTM layers with dropout for regularization and dense layers for regression output, optimized for time-series data.

GRU Model — Architecture & Key Outcomes

Architecture & Training

A sequential GRU model optimized for time-series RUL prediction:

- 3 Stacked GRU Layers: 128 → 64 → 32 units
- Dropout: 0.3 for regularization
- Dense Head: Single output neuron
- Optimizer: Adam, Loss: MSE, Early Stopping

Results & Advantages

Robust performance on the unseen test dataset:

- RMSE: 13.80
- R^2: 0.89
- MAE: 10.33
- NASA Score: 299.28

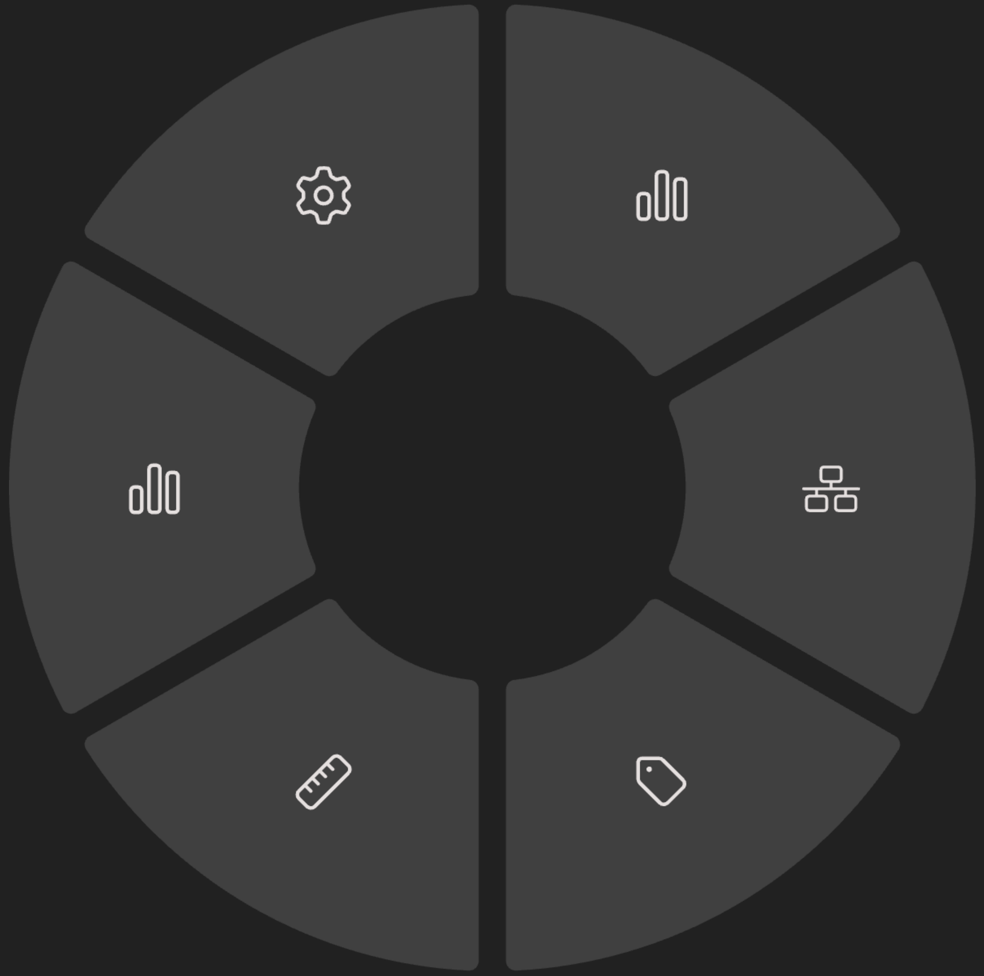
The GRU model provides competitive accuracy with reduced computational overhead, making it suitable for practical deployment.

Training Configuration & Evaluation

Rigorous training configurations and comprehensive evaluation metrics were applied to ensure the robustness and accuracy of our predictive models.

Classification Training

80/20 train-test split with class imbalance addressed for improved performance on rare failure classes.



LSTM Training

Trained with EarlyStopping (100 epochs, batch size 32, 20% validation split) to prevent overfitting.

GRU Training

Utilized Adam optimizer and Mean Squared Error (MSE) loss, with Early Stopping to prevent overfitting. Architecture includes 3 stacked GRU layers (128→64→32 units).

Classification Evaluation

Assessed using Accuracy, Precision, Recall, and F1-Score, focusing on the "Failure" class.

LSTM Evaluation

Evaluated with RMSE, R^2, and the NASA C-MAPSS scoring function, standard for RUL prediction.

GRU Evaluation

Metrics include RMSE: 13.80, R^2: 0.89, MAE: 10.33, and NASA Score: 299.28, demonstrating robust RUL prediction.

A detailed analysis and comparison of all model performances can be found in a separate file: `Model_Comparison.md`. Finalized model export and deployment are currently in progress.

Team Members – Team 10

Meet the dedicated individuals behind the predictive maintenance project, each contributing their expertise to deliver a robust and insightful solution.

1

Abdelrahman Sobhy

Role: Deployment & Model Training

- Built and optimized the LSTM Deep Learning model
- Web deployment

2

Abdallah Mousa

Role: Model Training & Validation

- ML models (Random Forest, Gradient Boosting Classifier, SVM.)
- Documentations (README, Reports, project presentation)

3

Abdelrahman Ghoraba

Role: Exploratory Data Analysis (EDA)

- Feature Engineering (RUL + Rolling Features)
- Statistical insights

4

Mohamed Gamal Zain Elabdein

Role: Data Preparation & Model training & Validation

- Raw data processing
- Build model with GRU Layer

5

Abdalla Ahmed Abuelhadid

Role: Exploratory Data Analysis (EDA)

- Data cleaning & Visualization
- Documentations (README, Reports)

6

Mohamed Khaled Yahya

Role: Data Preparation & Model training & Validation

- Raw data processing
- Build model with GRU Layer

7

Abdullah AbdalGawad

Role: Data Preparation

- dataset handling and formatting