Configuration Manual

MSc Research Project
Data Analytics

Khamalesh Ramesh
Student ID: x23325216

School of Computing
National College of Ireland

Supervisor: Mr. Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Khamalesh Ramesh

Student ID: x23325216

Programme: Data Analytics **Year:** 2024-2025

Module: Research Practicum 2

Supervisor: Mr. Hicham Rifai

Submission Due Date: Monday, 11 August 2025

Project Title: Configuration Manual

Word Count: XXX

Page Count: 5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Khamalesh Ramesh

Date: Monday, 11 August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Khamalesh Ramesh
x23325216

1. Introduction
This configuration manual provides step-by-step instructions to replicate the NASA Turbofan Engine Remaining Useful Life (RUL) Prediction research project. It covers the hardware, software, dataset setup, and execution process required for full reproducibility.

The project applies deep learning techniques to the NASA C-MAPSS dataset for predictive maintenance in aerospace systems. A hybrid CNN–BiLSTM–Attention model is used along with wavelet-based denoising, statistical feature extraction, and Optuna hyperparameter tuning.

By following the instructions, users will be able to:

1. Set up the required hardware and software.
2. Prepare and preprocess the dataset.
3. Train, tune, and evaluate the model.
4. Reproduce results, metrics, and visualizations.

2. System Specifications
2.1 Hardware Configuration

- Device: Apple MacBook Air (M2, 2022)
- Operating System: macOS Sequoia 15.5 (64-bit)
- Processor: Apple M2 Chip
- Memory (RAM): 8 GB
- Storage: 245.11 GB SSD

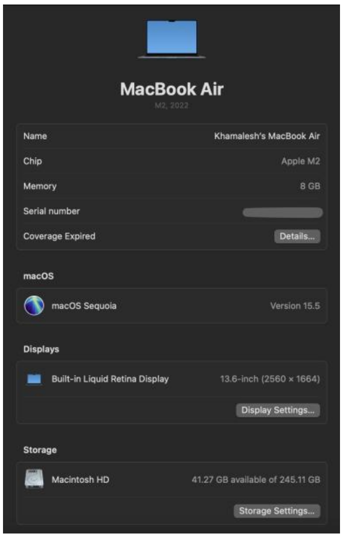

Figure1: Screenshot of "About This Mac" system information

2.2 Software and Tools

- Python: 3.10+
- PyTorch: 2.0+ with Apple MPS backend
- Development Environments: Visual Studio Code, JupyterLab
- Optimization Framework: Optuna (for hyperparameter tuning)
- Visualization Libraries: Matplotlib, Seaborn

3. Project Structure

The project is organized as follows:

```
├── checkpoints            # Saved model weights for each dataset
│   ├── best_model_FD001.pth
│   ├── best_model_FD002.pth
│   ├── best_model_FD003.pth
│   └── best_model_FD004.pth
├── data                   # CMAPSS dataset files & reference docs
│   ├── Damage Propagation Modeling.pdf
│   ├── readme.txt
│   ├── RUL_FD001.txt
│   ├── RUL_FD002.txt
│   ├── RUL_FD003.txt
│   ├── RUL_FD004.txt
│   ├── test_FD001.txt
│   ├── test_FD002.txt
│   ├── test_FD003.txt
│   ├── test_FD004.txt
│   ├── train_FD001.txt
│   ├── train_FD002.txt
│   ├── train_FD003.txt
│   └── train_FD004.txt
├── experiments            # Training & tuning scripts
│   ├── final_retrain_evaluate.py
│   └── optuna_tuning_per_dataset.py
├── loss_logs              # Per-epoch training and validation loss records
│   ├── FD001_losses.csv
│   ├── FD002_losses.csv
│   ├── FD003_losses.csv
│   └── FD004_losses.csv
├── plots                  # Model evaluation plots & summaries
│   ├── actual_vs_predicted_FD001.png
│   ├── actual_vs_predicted_FD002.png
│   ├── actual_vs_predicted_FD003.png
│   ├── actual_vs_predicted_FD004.png
│   ├── final_model_summary.csv
│   ├── residuals_FD001.png
│   ├── residuals_FD002.png
│   ├── residuals_FD003.png
│   └── residuals_FD004.png
├── README.txt             # Project overview and basic instructions
├── requirements.txt       # Python dependencies
├── results                # Final prediction results & summaries
│   └── rul_results_summary.xlsx
├── snapshots              # Preprocessed data snapshots
│   └── df_snapshots
└── src                    # Core source code modules
    ├── data_loader.py          # Data loading, denoising, labeling, feature extraction
    ├── dataset.py              # Sequence generation and dataset formatting
    ├── evaluate.py             # Model evaluation and visualization
    ├── losses.py               # Loss function definitions
    ├── model.py                # CNN–BiLSTM–Attention architecture
    ├── preprocess_combined.py  # Combined preprocessing steps
    └── train.py                # Training loop, early stopping, checkpointing
```

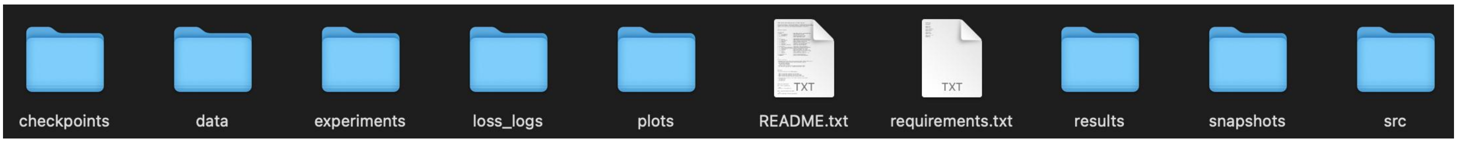

Figure 2: Screenshot of Finder view showing the rul-prediction folder structure

4. Python Environment & Dependencies

All dependencies were installed system-wide via pip without using a virtual environment.
The required Python packages are listed in requirements.txt:

1. torch>=2.0.0
2. torchvision
3. torchaudio
4. numpy>=1.22
5. pandas>=1.5
6. scikit-learn>=1.1
7. matplotlib>=3.6
8. seaborn>=0.12
9. optuna>=3.0
10. openpyxl>=3.0
11. PyWavelets>=1.4.0
12. tqdm>=4.64
13. joblib>=1.2

```
(base) khamaleshramesh@Khamaleshs-MacBook-Air rul-prediction % pip install -r requirements.txt

Requirement already satisfied: torch>=2.0.0 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 1)) (2.7.0)
Requirement already satisfied: torchvision in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 2)) (0.22.0)
Requirement already satisfied: torchaudio in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 3)) (2.7.0)
Requirement already satisfied: numpy>=1.22 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 5)) (1.26.4)
Requirement already satisfied: pandas>=1.5 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 6)) (2.2.2)
Requirement already satisfied: scikit-learn>=1.1 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 8)) (1.4.2)
Requirement already satisfied: matplotlib>=3.6 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 9)) (3.8.4)
Requirement already satisfied: seaborn>=0.12 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 10)) (0.13.2)
Requirement already satisfied: optuna>=3.0 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 12)) (4.3.0)
Requirement already satisfied: openpyxl>=3.0 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 13)) (3.1.2)
Requirement already satisfied: PyWavelets>=1.4.0 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 15)) (1.5.0)
Requirement already satisfied: tqdm>=4.64 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 16)) (4.66.4)
Requirement already satisfied: joblib>=1.2 in /opt/anaconda3/lib/python3.12/site-packages (from -r requirements.txt (line 17)) (1.4.2)
Requirement already satisfied: filelock in /opt/anaconda3/lib/python3.12/site-packages (from torch>=2.0.0->-r requirements.txt (line 1)) (3.13.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /opt/anaconda3/lib/python3.12/site-packages (from torch>=2.0.0->-r requirements.txt (line 1)) (4.12.2)
Requirement already satisfied: setuptools in /opt/anaconda3/lib/python3.12/site-packages (from torch>=2.0.0->-r requirements.txt (line 1)) (69.5.1)
Requirement already satisfied: sympy>=1.13.3 in /opt/anaconda3/lib/python3.12/site-packages (from torch>=2.0.0->-r requirements.txt (line 1)) (1.14.0)
Requirement already satisfied: networkx in /opt/anaconda3/lib/python3.12/site-packages (from torch>=2.0.0->-r requirements.txt (line 1)) (3.2.1)
Requirement already satisfied: jinja2 in /opt/anaconda3/lib/python3.12/site-packages (from torch>=2.0.0->-r requirements.txt (line 1)) (3.1.4)
Requirement already satisfied: fsspec in /opt/anaconda3/lib/python3.12/site-packages (from torch>=2.0.0->-r requirements.txt (line 1)) (2025.3.0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /opt/anaconda3/lib/python3.12/site-packages (from torchvision->-r requirements.txt (line 2)) (10.3.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/anaconda3/lib/python3.12/site-packages (from pandas>=1.5->-r requirements.txt (line 6)) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.12/site-packages (from pandas>=1.5->-r requirements.txt (line 6)) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/lib/python3.12/site-packages (from pandas>=1.5->-r requirements.txt (line 6)) (2023.3)
Requirement already satisfied: scipy>=1.6.0 in /opt/anaconda3/lib/python3.12/site-packages (from scikit-learn>=1.1->-r requirements.txt (line 8)) (1.15.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/anaconda3/lib/python3.12/site-packages (from scikit-learn>=1.1->-r requirements.txt (line 8)) (2.2.0)
Requirement already satisfied: contourpy>=1.0.1 in /opt/anaconda3/lib/python3.12/site-packages (from matplotlib>=3.6->-r requirements.txt (line 9)) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /opt/anaconda3/lib/python3.12/site-packages (from matplotlib>=3.6->-r requirements.txt (line 9)) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /opt/anaconda3/lib/python3.12/site-packages (from matplotlib>=3.6->-r requirements.txt (line 9)) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/anaconda3/lib/python3.12/site-packages (from matplotlib>=3.6->-r requirements.txt (line 9)) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/lib/python3.12/site-packages (from matplotlib>=3.6->-r requirements.txt (line 9)) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/anaconda3/lib/python3.12/site-packages (from matplotlib>=3.6->-r requirements.txt (line 9)) (3.0.9)
Requirement already satisfied: alembic>=1.5.0 in /opt/anaconda3/lib/python3.12/site-packages (from optuna>=3.0->-r requirements.txt (line 12)) (1.15.1)
Requirement already satisfied: colorlog in /opt/anaconda3/lib/python3.12/site-packages (from optuna>=3.0->-r requirements.txt (line 12)) (6.9.0)
Requirement already satisfied: sqlalchemy>=1.4.2 in /opt/anaconda3/lib/python3.12/site-packages (from optuna>=3.0->-r requirements.txt (line 12)) (1.4.54)
Requirement already satisfied: PyYAML in /opt/anaconda3/lib/python3.12/site-packages (from optuna>=3.0->-r requirements.txt (line 12)) (6.0.1)
Requirement already satisfied: et-xmlfile in /opt/anaconda3/lib/python3.12/site-packages (from openpyxl>=3.0->-r requirements.txt (line 13)) (1.1.0)
Requirement already satisfied: Mako in /opt/anaconda3/lib/python3.12/site-packages (from alembic>=1.5.0->optuna>=3.0->-r requirements.txt (line 12)) (1.3.9)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.12/site-packages (from python-dateutil>=2.8.2->pandas>=1.5->-r requirements.txt (line 6)) (1.16.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /opt/anaconda3/lib/python3.12/site-packages (from sympy>=1.13.3->torch>=2.0.0->-r requirements.txt (line 1)) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/anaconda3/lib/python3.12/site-packages (from jinja2->torch>=2.0.0->-r requirements.txt (line 1)) (2.1.3)
```

Figure 3: Screenshot of terminal showing pip install -r requirements.txt execution

5. Dataset Information

- Dataset Name: NASA C-MAPSS (FD001–FD004)
- Type: Engine degradation time-series dataset
- Total Size: ~21 MB (combined)
- Source: https://data.nasa.gov/dataset/C-MAPSS/

Each subset represents different operating conditions and fault modes for turbofan engines:

- FD001: Single operating condition, single fault mode
- FD002: Multiple operating conditions, single fault mode
- FD003: Single operating condition, multiple fault modes

- FD004: Multiple operating conditions, multiple fault modes

```
(base) khamaleshramesh@Khamaleshs-MacBook-Air data % ls -lh

total 88648
-rw-r--r--@ 1 khamaleshramesh  staff   424K  3 Oct  2008 Damage Propagation Modeling.pdf
-rw-r--r--@ 1 khamaleshramesh  staff   2.4K  3 Oct  2008 readme.txt
-rw-r--r--@ 1 khamaleshramesh  staff   429B  1 Oct  2008 RUL_FD001.txt
-rw-r--r--@ 1 khamaleshramesh  staff   1.1K  1 Oct  2008 RUL_FD002.txt
-rw-r--r--@ 1 khamaleshramesh  staff   428B  1 Oct  2008 RUL_FD003.txt
-rw-r--r--@ 1 khamaleshramesh  staff   1.1K  1 Oct  2008 RUL_FD004.txt
-rw-r--r--@ 1 khamaleshramesh  staff   2.1M 25 Mar  2008 test_FD001.txt
-rw-r--r--@ 1 khamaleshramesh  staff   5.5M 17 Sep  2008 test_FD002.txt
-rw-r--r--@ 1 khamaleshramesh  staff   2.7M 25 Mar  2008 test_FD003.txt
-rw-r--r--@ 1 khamaleshramesh  staff   6.6M 17 Sep  2008 test_FD004.txt
-rw-r--r--@ 1 khamaleshramesh  staff   3.4M 25 Mar  2008 train_FD001.txt
-rw-r--r--@ 1 khamaleshramesh  staff   8.7M 17 Sep  2008 train_FD002.txt
-rw-r--r--@ 1 khamaleshramesh  staff   4.0M 25 Mar  2008 train_FD003.txt
-rw-r--r--@ 1 khamaleshramesh  staff   9.9M 17 Sep  2008 train_FD004.txt
```

Figure 4: Screenshot of data folder showing dataset files

6. Module Overview

6.1 src/data_loader.py

- Reads CMAPSS dataset files.
- Applies wavelet-based denoising to sensor signals.
- Labels data with Remaining Useful Life (RUL).
- Performs statistical feature extraction.

6.2 src/dataset.py

- Converts data into supervised learning format.
- Handles sequence generation for time-series inputs.

6.3 src/model.py

- Defines the CNN + BiLSTM + Attention architecture.
- Outputs scalar RUL predictions.

6.4 src/train.py

- Implements training loop with early stopping and checkpoint saving.
- Uses composite loss function (MSE + Huber).

6.5 src/evaluate.py

- Computes performance metrics: RMSE, MAE, R^2.
- Plots attention weights, RUL trajectories, and residuals.

7. Hyperparameter Optimization

- Script Used: experiments/optuna_tuning_per_dataset.py
- Objective: Minimize validation RMSE for each CMAPSS subset.
- Number of Trials: 50 per dataset.

Search Space:

1. Learning rate (learning_rate)
2. Batch size (batch_size)
3. Sequence length (sequence_length)
4. Dropout rate (dropout_rate)
5. Hidden size (hidden_size)

8. Execution Flow

8.1 Run Training and Evaluation

To train and evaluate the model on a selected dataset:

python experiments/final_retrain_evaluate.py

8.2 Run Hyperparameter Tuning
To run Optuna-based hyperparameter tuning for a specific dataset :

python experiments/optuna_tuning_per_dataset.py

8.3 Notes
- Ensure the dataset files are placed in the data/ directory.
- Output results, plots, and logs will be stored automatically in their respective folders.

9. Outputs
After execution, the following output directories are generated:
- /results/ – Final prediction files (.csv) for each dataset.
- /plots/ – Visualization outputs, including:
  - RUL vs Actual plots
  - Residual plots
- /checkpoints/ – Saved model weights for each dataset.
- /logs/ – Training logs, loss curves, and evaluation metrics.

10. Conclusion
This configuration manual provides all the technical details required to replicate the NASA Turbofan Engine Remaining Useful Life (RUL) Prediction project.

It includes hardware and software specifications, project structure, dataset preparation, module descriptions, hyperparameter optimization settings, execution commands, and output formats.

Following these steps ensures that the experimental results can be reproduced consistently for academic validation, industrial application, or further research in predictive maintenance for aerospace systems.