Project Code Structure & Execution Guide

This document explains **how the codebase is structured**, **which scripts to run**, and **where to find results**. It is intended as a **practical guide**, not a detailed methodological explanation.

---

1) General Structure (How everything is launched)

Purpose

Centralize the execution of experiments (models × configurations) and post-processing of results.

Main entry point

Script to run - `main.py`

How to run

```
python main.py
```

What to configure before running

In `main.py`, you should: - Set the path to a **prepared dataset** (already cleaned): - Example: `Data_debit/clean_data_individual/ARB_prepared.csv` - Select which configurations to run (e.g. `A`, `B`): - `config_list = ["A", "B"]` - Uncomment the model(s) you want to execute (Tree, XGBoost, LSTM, GRU)

Where to find results

After execution, results are written to: - `Results/Config_<Config>/individual_results/merged_<type>/` - A global summary file: - `Results/horizon_effect_summary.csv`

You can inspect the results by opening the generated CSV files.

---

2) Machine Learning Scripts (Models & Evaluation)

These scripts **train, evaluate, and export results** for each model. They are **not meant to be run directly** in normal usage. Instead, they are called from `main.py`.

All ML scripts rely on shared utilities in: - `ML_function.py` (data loading, feature selection by config, windowing, hyperparameter search, evaluation)

They all expect a **prepared station-level CSV file** as input.

2.1 Tree-based model

File - `tree_model.py`

How to run - Uncomment the corresponding call in `main.py` - Run:

```
python main.py
```

Outputs - CSV files such as: - `results_Tree.csv` - `per_horizon_Tree.csv` - Test equivalents by configuration

---

2.2 XGBoost model

File - `XGBoost.py`

How to run - Uncomment the XGBoost call in `main.py` - Run:

```
python main.py
```

Outputs - CSV files such as: - `results_XGBoost.csv` - `per_horizon_XGBoost.csv` - Test equivalents by configuration

---

2.3 LSTM model

File - `LSTM.py`

How to run - Uncomment the LSTM call in `main.py` - Run:

```
python main.py
```

Outputs - CSV files such as: - `results_LSTM.csv` - `per_horizon_LSTM.csv` - Test equivalents by configuration

---

2.4 GRU model

File - `GRU.py`

How to run - Uncomment the GRU call in `main.py` - Run:

```
python main.py
```

Outputs - CSV files such as: - `results_GRU.csv` - `per_horizon_GRU.csv` - Test equivalents by configuration

---

2.5 Result merging & statistical summaries

Files - `result_table_analyse.py` - `Stat_function.py`

How to run - These scripts are executed automatically at the end of `main.py`

Outputs - Merged result tables per configuration in: - `Results/Config_<Config>/individual_results/merged_<type>/` - Global horizon-effect summary: - `Results/horizon_effect_summary.csv`

---

3) Data Scripts (`data*`) — Data Preparation Pipeline

Important recommendation

All scripts whose name starts with `data...` are responsible for **data preparation** (cleaning, aggregation, feature construction).

👉**We recommend NOT running these scripts by default.** - They can take significant time - They may require substantial disk space - The machine learning pipeline already uses **prepared and cleaned datasets**

You can safely run the ML experiments **without executing any data scripts**, as long as prepared CSV files are available.

When to run data scripts

You may choose to run `data*` scripts if you want to: - Rebuild the prepared datasets from raw data - Modify the data cleaning process - Regenerate engineered features (rolling means, sums, etc.)

How to run data scripts

Each data script can be run individually:

```
python data_<script_name>.py
```

Where data outputs appear

Data scripts typically generate or overwrite files in: - `Data_debit/clean_data_individual/`

Once data preparation is finished, you can proceed with:

```
python main.py
```

Summary (Recommended Workflow)

1. **Do not run data scripts** unless you explicitly want to rebuild the data
2. Verify that a prepared `<STATION>_prepared.csv` file exists
3. Configure `main.py` (dataset path, configs, models)
4. Run:

   ```
   python main.py
   ```

5. Inspect results in the `Results/` directory