National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Ankith Babu Joseph
Student ID: 23185813

School of Computing
National College of Ireland

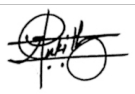
Supervisor: Harshani Nagahamulla

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ankith Babu Joseph
Student ID:	23185813
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Harshani Nagahamulla
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	333
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:	
Date:	12th December 2024

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

Ankith Babu Joseph
23185813

1 Prelude

This document outlines the steps to configure and run the Variational Mode Decomposition (VMD) and Fourier Analysis Network (FAN) hybrid model for forecasting Air Quality Index (AQI) time-series data. The manual includes details on the software environment, parameters, execution, and troubleshooting.

2 System Requirements

2.1 Hardware Requirements

- GPU: NVIDIA Tesla T4 or equivalent
- CPU: Intel Xeon @ 2.00GHz or higher
- RAM: Minimum 16 GB (51 GB recommended)
- Disk Space: Minimum 50 GB (256 GB recommended)

2.2 Software Requirements

- Python 3.10 or higher
- PyTorch 2.5.1+cu121
- Key Libraries: sktime, statsmodels, numpy, pandas, matplotlib, joblib

3 Installation

3.1 Using Google Colab

1. Open the google colab[1] and start a new notebook
2. Google colab kernel already is pre installed with many required libraries, to install the remaining run the pip install command

[1]Google Colab: `https://colab.google/`

```
%%capture
!pip install sktime
!pip install dask[dataframe]
```

3. Use git to clone the code files

```
!git clone https://github.com/ankithbjoseph/vmd-fan.git
```

4. go to the vmd-fan folder

```
cd vmd-fan
```

5. The required environment is setup and can continue with executing the code.

3.2 Or Install Locally

1. Install Python and create a virtual environment:

```
python3 -m venv venv
source venv/bin/activate  # For Linux/Mac
venv\Scripts\activate     # For Windows
```

2. Run the following command to install required dependencies:

```
pip install -r requirements.txt
```

3. Use git to clone the code files

```
!git clone https://github.com/ankithbjoseph/vmd-fan.git
```

4 Configuration

Parameters

All parameters are specified in a JSON file (`parameters.json`) with the following structure:

Listing 1: parameters.json

```
{
  "VMD": {
      "K": [3,5,7],
      "alpha": [1000,3000],
```

```
        "tau": [1e-6],
        "DC": [false],
        "tol": [1e-5,1e-7]
    },
    "FAN": {
        "window_size": [12],
        "learning_rate": [0.0001,0.00001],
        "p_ratio": [0.15,0.25],
        "fan_units": [64]
    },
    "BaseNN": {
        "window_size": [12],
        "learning_rate": [0.0001,0.00001]
    }
}
```

Command-Line Arguments

The script `main.py` accepts the following arguments:

Argument	Description	Default Value
`--eda`	Perform Exploratory Data Analysis (EDA).	`False`
`--vmd`	Enable VMD decomposition.	`False`
`--input_file`	Path to the dataset file.	`./dataset/air_quality_Taiwan.csv`
`--variable`	Target variable for forecasting.	`aqi`
`--output_dir`	Directory to save results.	`results`
`--model`	Model to use (`FAN` or `BaseNN`).	`None`
`--visualize`	Save visualizations of results.	`False`
`--num_jobs`	Number of parallel jobs.	`1`
`--forecast_horizon`	Number of steps to forecast.	`1`
`--parameter_grid`	Path to parameter grid JSON.	`parameters.json`
`--cities`	List of cities to process (comma-separated).	`all cities in dataset`

Table 1: Command-Line Arguments for `main.py`

5 Code Execution

Step 1: Data Preprocessing

Run the script to clean and segment the data:

```
python main.py --cities "Annan" --eda --visualize --num_jobs 40
```

Step 2: VMD Decomposition

Decompose the dataset into Intrinsic Mode Functions (IMFs):

```
python main.py --cities "Annan" --vmd --visualize --num_jobs 40
```

Step 3: Model Training and Forecasting

Run the FAN or BaseNN model on the decomposed data:

```
python main.py --cities "Annan"  --model 'BaseNN' --visualize --num_jobs 40
```

```
python main.py --cities "Annan" --model 'FAN' --visualize --num_jobs 40
```

```
python main.py --cities "Annan" --vmd --model 'BaseNN' --visualize --num_jobs 40
```

```
python main.py --cities "Annan" --vmd --model 'FAN' --visualize --num_jobs 40
```

6 Directory Structure

The output directory will have the following structure:

```
results/
  |-- city_name/
       |-- city_name_aqi.csv
       |-- vmd_decompositions/
       |-- plots/
  |-- results.csv
```

7 Troubleshooting

Parameter Errors

- Verify the syntax of `parameters.json`. Ensure it is valid JSON.
- Check that the parameter values align with the model requirements.

CUDA Errors

- Ensure that PyTorch is installed with CUDA support. Install the correct version by visiting the PyTorch Installation Guide[2].
- Check GPU availability using the following Python command:

```
import torch
print(torch.cuda.is_available())
```

- If `torch.cuda.is_available()` returns `False`, verify that:
  - The system has a compatible NVIDIA GPU.
  - NVIDIA drivers and CUDA toolkit are installed and up-to-date.

[2]PyTorch Installation Guide: `https://pytorch.org/get-started/locally/`