National College of Ireland

Configuration Manual

MSc Research Project
MSCDAD_A_JAN25I

Manjima Panangath Manoj
Student ID: 23380543

School of Computing
National College of Ireland

Supervisor: Shubham Subhnil

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Manjima Panangath Manoj ……

Student ID: ……23380543…… ….……

Programme: …… MSCDAD_A_JAN25I …… **Year:** 2025 ……..

Module: ……Research Practicum ……

Lecturer: ……Shubham Subhnil ……

Submission Due Date: 28/01/2026…… ….……

Project Title: Evaluation of Urban Noise Mitigation Measures Using Sonitus Sensor Data: A Case Study of Dublin ……

Word Count: …… **Page Count:** ……

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: ……Manjima Panangath Manoj ……

Date: …27/01/2026…… ……

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

Manjima Panangath Manoj
Student ID: 23380543

1 Introduction

This configuration document reports the entire computational environment, software environment, system configuration and the analysis workflow in the current research project investigating the effectiveness of the noise mitigation measures in Dublin using Sonitus sensor measurements. The manual includes specifications of the hardware and operating system used, and the development environment, structuring and preparation of input datasets, and the processing pipeline, which is reproducible in Python. This manual is meant to make sure that the analytical process and model implementation and visual products created in the course of the research can be recreated exactly with the same configuration parameters and file formatting. Visual Studio Code (VS Code) and Jupyter Notebook have been used to do all analyses alongside modern scientific Python libraries.

2 System Overview

All the computational work was done on an MSI GF63 Thin 10SCSR laptop with an Intel 10750H processor (2.60 GHz, 6 cores and 12 threads), 8 GB of RAM (7.83 usable) and a Windows 11 Home Single Language operating system (Version 25H2, OS Build 26200.7462) 64-bit. The graphics processor of the device is an NVIDIA GeForce GTX 1650 Max-Q, but in this research, there was no need to use a GPU acceleration. This system had enough computing power to run Python scripts, perform regression and Difference-in-Differences modelling, create graphical outputs and process large, merged noise and weather datasets. The system has worked well during the project and its setup is sufficient to carry out all the analytical duties given in this manual.
The main integrated development environment (IDE) was Visual Studio Code (VS Code) which was used to develop and execute the project. VS code offered an effective, modular, and extendable platform to run Jupyter notebooks, Python programs, and data visualisation endeavours with the Python and Jupyter extensions. Python 3.9 (or above) was the basic programming language

Component	**Specification**
Device	MSI GF63 Thin 10SCSR
Processor	Intel® Core™ i7-10750H @ 2.60 GHz (6 cores, 12 threads)
Installed RAM	8 GB (7.83 GB usable)
System Type	64-bit OS, x64-based processor
GPU	NVIDIA GeForce GTX 1650 Max-Q
OS Version	Windows 11 Home Single Language, Version 25H2
OS Build	26200.7462

Component	Specification
Installation Date	27-10-2024

Table: 1

3 Software Environment

The analytical process was done using a collection of scientific libraries that was a requisite of numerical calculation, statistical modelling and visualisation. Such libraries are pandas to manipulate structured data, NumPy to perform numerical computations, matplotlib and seaborn to create figures, stats models to do OLS and Difference-in-Differences estimations, and scikit-learn to perform Random Forest regression and feature importance analysis. It is fully reproducible and is in line with the best practices in environmental data science.
The project was designed to have a clear modular directory structure to allow managing the raw data, processed dataset, notebooks, and outputs. The hierarchy below is the one that is reflected in the project explorer in the Visual Studio Code.
Such a folder structure is reproducible as it clearly separates raw inputs, processing scripts, modelling notebooks and output artefacts.

Software / Library	Version	Purpose
Visual Studio Code	Latest	Development and execution environment
Python	3.9+	Primary programming language
pandas	1.5+	Data cleaning and manipulation
NumPy	1.23+	Numerical operations
seaborn	0.12+	Statistical visualisation
matplotlib	3.5+	Plotting
statsmodels	0.14+	Regression and DID modelling
scikit-learn	1.2+	Random Forest modelling
Jupyter Extension	Latest	Notebook execution in VS Code

Table: 2

4 Directory Structure

To facilitate a clear management of raw data, processed data, notebooks and outputs, the project was organised in a clear modular directory structure. The hierarchy below is an indication of the arrangement that is apparent in the project explorer of Visual Studio Code.

Folder	Contents	Purpose
data_raw/	Sonitus noise CSVs, weather files, KML intervention files, metadata	Stores all raw, unprocessed datasets used for analysis
notebooks/	noise_mitigation.ipynb, data_collection.ipynb	Contains analytical workflow, preprocessing, modelling and visualisation code
outputs/	residual plots, regression figures, DiD plots, summary tables	Stores all generated analysis outputs and graphics

Folder	Contents	Purpose
documents/	MSc Research Report, Presentation, Configuration Manual	Maintains all written deliverables and final documentation

Table:3

5 Data Inputs

Three major classes of data were used in the research:

- Noise Data: Hourly LAeq values of Sonitus monitoring stations.
- Meteorological Data: The temperature, humidity, speed of wind, and the amount of rain were taken to Met Éireann.
- Intervention Metadata: Documents of intervention type, location, date of implementation.

The noise data files were supplied in CSV format and had to be preprocessed, to provide consistency in the format of timestamps, missing values, and column names. The weather data were collected and combined with noise values on a common index of the datetimes. The intervention layers that are stored in KML format were analysed and assigned control vs treated labels to the monitoring sites.

6 Data Processing and Analytical Pipeline

The processing pipeline of the data was reproducible as it was based on a multi-stage workflow. Raw noise and weather data were first imported into Python and cleaned by removing invalid or missing values and standardized. These refined datasets were combined with one analysis file which comprised hourly LAeq and related meteorological variables. Noise indicators which were daily and END-compliant (LDEN and LNIGHT) were calculated with the Directive 2002/49/EC formulae. After processing, the data set was ready to be analyzed statistically. The baseline patterns were studied using descriptive statistics, and then hypothesis testing, Difference-in-Differences estimation, and regression modelling. Seaborn and matplotlib were used to generate all the plots such as pre-post distributions, actual vs predicted scatterplots, and residual diagnostics and were automatically saved in the output's directory.

7 Execution Procedure

Step 1 - Open the project in Visual Studio Code

1. Launch the Visual Studio Code
2. Use File-Open Folder and select the main Project Folder (Manjima_Dessertation)

Step 2 - Activate the Python Environment

1. Ensure Python 3.9+ is installed on the machine.
2. Create and activate the Virtual environment in the Project folder.
3. Install all required libraries listed in Table:2 using pip (e.g. pip install pandas NumPy matplotlib seaborn statsmodels scikit-learn).

Step 3 – Open the Main Notebook

1. In the VS Code Explorer, navigate to the notebooks folder.
2. Open noise_mitigation.ipynb.

3. Select the correct Python interpreter/ virtual environment when prompted.

Step 4 – Run the Notebook cells

1. Run the cells from top to bottom using Run All or by executing cell by cell
2. The notebook will automatically:
    - Import and clean the raw noise, weather and intervention data:
    - Compute LDEN and LNIGHT indicators
    - Assign treated and control sites
    - Apply descriptive statistics, Difference-in-Differences and Regression models.

Step 5 – Verify Outputs

1. Check that figures and tables are generated without errors
2. Confirm that output files (Plots, Summary tables, model summaries) are saved into the output's directory.

Steps 6 – Reproducibility

1. The notebook is fully annotated and can be rerun without modification on any machine where the software listed in Table:2 is installed.
2. To repeat the analysis, the user only needs the same folder structure, raw data files in data raw, and the required Python Libraries

8 Troubleshooting and Reproducibility

Missing files of data, import problems, or mismatched paths are common problems that are solved by ensuring that the folder structure is correct and that all the dependencies are installed. Wrong formats of date time can lead to merging errors; they are solved by examination of parsing functions of timestamps. Random Forest procedures must be reproducible so that random seeds are fixed. As much as version control is not a mandatory aspect, Git is suggested when one wants to record a transparent update history.

References