National College of Ireland

Configuration Manual

MSc Research Project
Programme Data Analytics

Sanket Sawant
Student ID: x23268077

School of Computing
National College of Ireland

Supervisor: Prof. Vikas Tomer

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sanket Sawant
Student ID:	X23268077
Programme:	MSc Data Analytics **Year:** 2024 2025
Module:	Research Practicum
Lecturer:	Prof. Vikas Tomer
Submission Due Date:	15th September 2025
Project Title:	Explainable Stacked Ensemble Learning for accurate wind power forecasting using Scada time series data
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: Sanket Sawant

Date: 15th September 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: Wind Turbine Power Generation Prediction

---

1. Introduction

1.1 Background

The increasing reliance on renewable energy sources such as wind energy necessitates accurate power generation forecasting to optimize energy production and grid stability. Wind turbine output is inherently variable due to changing weather conditions, making predictive modeling a critical aspect of wind farm management.

This project utilizes machine learning models to predict power output from wind turbines using real-time weather data. By leveraging historical data, the models can learn patterns that help forecast future power output. The dataset used is publicly available from Mendeley Data and contains wind speed, direction, and other related variables collected from a wind turbine system.

1.2 Aim of the Study

The objective of this project is to develop and evaluate regression models to accurately predict wind turbine power generation, compare multiple models, and interpret model predictions using explainable AI (XAI) techniques.

1.3 Research Objectives

1. To preprocess and normalize the wind turbine dataset.
2. To perform exploratory data analysis to understand feature relationships.
3. To engineer rolling window-based features suitable for time-series regression.
4. To develop and evaluate five machine learning models:
    - Support Vector Regressor (SVR)
    - Random Forest Regressor
    - XGBoost
    - AdaBoost
    - Stacking Regressor (Ensemble)
5. To compare model performance using evaluation metrics like MAE and RMSE.
6. To apply Explainable AI techniques (e.g., SHAP or LIME) for model interpretability.
7. To forecast future wind turbine power output using the best-performing model.

1.4 Research Question

How does a stacked ensemble model combining multiple machine learning algorithms improve the accuracy and robustness of wind power forecasting using SCADA data, and which SCADA features most significantly influence the predictions as revealed by explainable AI techniques (LIME)?

---

2. Environment Setup

2.1 Development Platform

- **Platform**: Kaggle Notebooks
- **Advantages**: Free access to GPUs and TPUs, extensive datasets, and built-in library support.

2.2 Hardware (Provided by Kaggle)

- 13 GB RAM
- Dual-core CPU
- Tesla T4 GPU (optional for faster training)
- 20 GB disk space

2.3 Software and Tools

Tool	Description
Python 3.x	Programming language
Pandas	Data loading and manipulation
Numpy	Numerical computing
Sklearn	ML models, metrics, preprocessing
XGBoost	Boosted tree regression model
SHAP/LIME	Model explainability
Seaborn	Statistical data visualization
Matplotlib	Plotting and visualization
Plotly	Interactive forecasting visualization

---

3. Dataset Description

Dataset Source

- **Title**: Wind Turbine SCADA Dataset
- **Link**: Mendeley Data
- **Format**: CSV

Data Overview

The dataset consists of time-stamped environmental measurements and corresponding wind turbine power output.

- **Total Records**: ~50,000
- **Features** (example):
  - Wind Speed (m/s)
  - Wind Direction (°)
  - Temperature (°C)
  - Power (kW) (Target)
  - Date/Time

Preprocessing Steps

- Handled missing/null values.
- Normalized features using MinMaxScaler.
- Feature engineering using rolling windows.
- Time-aware train-test split to prevent data leakage.

---

4. Library Requirements

Library	Use Case
pandas	Data loading and frame manipulation
numpy	Numerical operations
matplotlib	Plotting residuals and actual vs predicted
seaborn	Correlation heatmap, pairplots
sklearn	ML models, metrics (MAE, RMSE), preprocessing
xgboost	XGBoost Regressor
lime or shap	Model interpretation and explainability
plotly	Interactive line plots for forecasted outputs

---

5. Project Workflow

Step 1: Data Loading

- Load CSV using pandas.read_csv().
- Parse timestamp for time-series analysis.

Step 2: Data Preprocessing

- Apply MinMaxScaler to normalize features.
- Handle outliers and nulls if present.

Step 3: Exploratory Data Analysis

- Visualize feature relationships using heatmaps, scatter plots.
- Analyze seasonal and temporal trends.

Step 4: Feature Engineering

- Apply rolling window to convert raw data into supervised learning format.

Step 5: Train-Test Split

- Split data into training and test sets using 80:20 ratio.
- Ensure chronological order for time-series prediction.

Step 6: Model Training

- Train the following models:
  - **SVR**: Captures non-linear relations.
  - **Random Forest Regressor**: Handles feature interactions well.
  - **XGBoost**: High-performance gradient boosting.
  - **AdaBoost**: Adaptive boosting for regression.
  - **Stacking Regressor**: Combines multiple models using a meta-learner.

Step 7: Evaluation

- Evaluate using:
  - **MAE (Mean Absolute Error)**
  - **RMSE (Root Mean Squared Error)**
- Compare results using bar plots.

Step 8: Explainable AI

- Use LIME to:
  - Identify key features driving predictions.
  - Visualize global and local explanations.

Step 9: Forecasting

- Use trained models to predict unseen/future data.
- Forecast upcoming wind power output.
- Plot actual vs forecasted using plotly.line().
