Progress Report 2

Data Pipeline Implementation & Exploratory Analysis

Project:	Physics-Informed Neural Networks for Wake-Aware Yaw and Blade Pitch Control
Student:	Ibrahim Alghrabi
Advisor:	Dr. Tariq Ahmed Albusyuni
Course:	MATH 619 | Term 252

Executive Summary

This report documents the completion of Phase 1 (Data & Environment Setup) as outlined in Progress Report 1. The primary accomplishments include: (1) development of a comprehensive data pipeline for wind turbine SCADA data, (2) exploratory data analysis using real operational data from the Kelmarsh Wind Farm (UK), (3) design of the PINN architecture with physics-informed loss functions, and (4) preparation of training-ready datasets. The Kelmarsh dataset contains 135,068 valid SCADA records across 6 Senvion MM92 turbines (2.05 MW each), covering January-July 2021.

1. Introduction

Following the implementation roadmap established in Progress Report 1, this reporting period focused on building the foundational data infrastructure necessary for PINN development. The Kelmarsh Wind Farm dataset was selected from the OpenWindSCADA collection due to its comprehensive measurements across 6 turbines with co-located meteorological data. The primary objectives were to: (a) establish a reproducible data pipeline for SCADA data processing, (b) conduct exploratory analysis to understand the relationships between control variables (yaw, pitch) and power output, and (c) design the neural network architecture that will incorporate physics constraints.

2. Data Source: Kelmarsh Wind Farm

The Kelmarsh Wind Farm is located in Northamptonshire, UK, and consists of 6 Senvion MM92 turbines, each rated at 2.05 MW with a 92m rotor diameter. The dataset, obtained from Zenodo, provides 10-minute averaged SCADA records including wind speed, wind direction, nacelle position (yaw), blade pitch angles, power output, and rotor speed. Data from January to July 2021 was used for this analysis.

3. Data Pipeline Development

A modular Python-based data pipeline was developed to handle the Kelmarsh SCADA data. The pipeline performs the following operations:

Data Loading: Reads CSV files for all 6 turbines, handling the custom header format and extracting relevant columns (wind speed, direction, power, yaw, pitch, rotor speed).

Data Cleaning: Removes NaN values in critical columns, filters to valid operating ranges (0-30 m/s wind, -50 to 2200 kW power), and excludes curtailed periods where power was artificially limited.

Feature Engineering: Computes yaw misalignment (difference between wind direction and nacelle position), classifies operating regions (below cut-in, Region 2, Region 3), and normalizes features for PINN training.

Table 1: Kelmarsh Dataset Statistics (After Cleaning)

Variable	Mean	Std Dev	Min	Max
Wind Speed (m/s)	5.91	2.96	0.13	22.42
Wind Direction (°)	184.9	100.5	0.0	360.0
Power (kW)	611.0	668.4	-17.8	2077.5
Yaw Angle (°)	186.0	100.8	0.0	360.0
Pitch Angle (°)	7.06	16.26	-170.1	92.5
Rotor Speed (RPM)	10.09	4.35	0.0	15.34

Total records: 135,068 across 6 turbines

4. Key Findings from Exploratory Analysis

Power Curve Behavior: The Kelmarsh turbines exhibit the expected three-region power curve with cut-in around 3 m/s and rated power (~2050 kW) achieved near 12 m/s. The binned power curve shows increasing variance at higher wind speeds, with some high-wind curtailment visible above 20 m/s.

Inter-Turbine Variation: Turbine 6 shows notably lower mean power (68.2% of maximum) compared to other turbines, suggesting possible wake effects or operational differences. Turbine 2 produces the highest average power. This variation motivates the need for wake-aware optimization.

Yaw Tracking: Yaw misalignment is approximately normally distributed with most values within ±20°. The power vs yaw analysis shows clear degradation: turbines produce ~1000 kW at 0-5° misalignment but only ~450 kW at 25-30° misalignment (in the 6-10 m/s wind range).

Pitch Behavior: Pitch angles cluster near 0° during Region 2 operation and increase during Region 3 for power regulation. Some anomalous pitch values (extreme negatives) indicate sensor issues or fault conditions that were filtered during cleaning.

Correlation Structure: Wind speed and power are strongly correlated (r=0.95). Yaw angle correlates well with wind direction (r=0.89). Notably, pitch angle shows negative correlation with power (r=-0.26) and strong negative correlation with rotor speed (r=-0.75), consistent with pitch-to-feather control in high winds.

5. PINN Architecture Design

The neural network architecture was designed based on successful implementations in the literature and will be implemented using the **pinnstorch** framework (Bafghi & Raissi, NeurIPS 2023), which provides PyTorch-based PINN implementations with built-in Navier-Stokes examples.

Table 2: PINN Architecture Specification

Component	Specification	Rationale
Input Layer	4 neurons	wind speed, direction, yaw, pitch
Hidden Layers	5 layers × 64 neurons	Balance depth/width for PDEs
Activation	Tanh	Smooth, twice differentiable for gradients
Output Layer	3 neurons	velocity (u, v), power
Framework	pinnstorch (PyTorch)	Navier-Stokes support, GPU acceleration

6. Challenges Encountered

Data Quality Issues: The Kelmarsh dataset contains some anomalous pitch angle readings (extreme negative values) and periods of curtailment. A cleaning pipeline was implemented to filter these while preserving valid operational data.

Framework Selection: After evaluating DeepXDE and NVIDIA Modulus, the decision was made to use **pinnstorch** due to its PyTorch foundation, active maintenance by Raissi's group (the original PINN authors), and built-in Navier-Stokes examples that can be adapted for wake modeling.

Coordinate System: Wind direction and yaw angle are in compass coordinates (0-360°), requiring careful handling of the wraparound at 0°/360° when computing misalignment angles.

7. Next Steps for Progress Report 3

The following tasks are planned for the next reporting period:

1. Install pinnstorch and run the Navier-Stokes example to validate the setup
2. Adapt the PINN architecture to accept yaw/pitch as control inputs
3. Train the PINN on the prepared Kelmarsh dataset
4. Compare PINN predictions against a baseline neural network (no physics)
5. Demonstrate gradient computation for yaw optimization

Appendix: Visualization figures and complete code available in the GitHub repository.