Statstical Modeling of Wave Energy Converters layout

Subject 8

Abensour, Assouly, Simatos

June 6, 2025

1. Introduction

The problem we here endorse to tackle is how to maximize the energy deilvered by waves through a wave energy farms. This problem is inspired by research detailed in the paper *Optimisation of Large Wave Farms Using a Multi-Strategy Evolutionary Framework.* The farms consist of numerous submerged Wave Energy Converters (WECs), each designed as spherical buoys tethered by three cables to a seabed-based power-take-off system.

The main objective of our project is to optimize the spatial layout of these converters to achieve the highest possible total energy capture for the entire farm. The engineering challenge lies in arranging tens to hundreds of buoys inside a rectangular site while *maximising* the annual average power P_Σ extracted from realistic, directionally spread sea states. Because every buoy re-radiates waves, the hydrodynamic interactions are strongly coupled and non-linear; a naïve search of layouts is therefore prohibitively expensive. A significant challenge lies in the complexity of hydrodynamic interactions between the multiple converters, making direct evaluations of each potential configuration expensive and time-consuming.

As finding the optimal layout of the WECs may seem straightforward, the challenge actually results in a particularly extensive and complex search space, characterized by high dimensionality given by the number of possible combinations. Therefore multimodality—numerous local optima configurations may exist, but these do not necessarily represent global optima. To overcome these difficulties, the study proposes a supervised machine learning approach that aims at finding the global optima of our equation.

2. Physical Problem

Before getting in the statistical modeling of the problem, we shall the physical perspective to the optimization equation. Our aim is to set buoys in the best possible configuration. Each buoy has three translational degrees of freedom (surge, sway, heave). After linearisation in the frequency domain and assembling all N bodies, the coupled dynamics read

$$(M + A)\,\ddot{\mathbf{X}} \;+\; \big(B + D_{\text{pto}}\big)\,\dot{\mathbf{X}} \;+\; K_{\text{pto}}\,\mathbf{X} \;=\; \mathbf{F}_{\text{exc}}, \tag{1}$$

where

- M — rigid-body mass matrix; A — added-mass matrix capturing fluid inertia,
- B — radiation damping; D_{pto} and K_{pto} — PTO damping & stiffness,
- $\mathbf{F}_{\text{exc}}$ — linearised wave-excitation force vector,
- $\mathbf{X} \in R^{3N}$ — vector of buoy displacements.

Power Capture

For a monochromatic wave of frequency ω from direction β, the absorbed power is

$$p(\omega, \beta) = \tfrac{1}{2}\,\dot{\mathbf{X}}^{*}\,D_{\text{pto}}\,\dot{\mathbf{X}}.$$

Integrating over the directional spectrum $S(\omega, \beta)$ of a sea state (H_s, T_p) and weighting by its occurrence probability $\mathcal{O}(H_s, T_p)$ yields

$$P(H_s, T_p) = \int_0^\infty \int_0^{2\pi} S(\omega, \beta)\, p(\omega, \beta)\, d\beta\, d\omega, \tag{2}$$

$$P_\Sigma = \sum_{H_s, T_p} P(H_s, T_p)\, \mathcal{O}(H_s, T_p). \tag{3}$$

3. From Physics to Data-Driven Surrogates

Evaluating (1)–(3) for a single layout involves computing frequency-domain hydrodynamics, solving large linear systems and integrating over hundreds of sea states—minutes to hours of CPU time. To enable fast optimisation, control, or uncertainty studies, we seek a *supervised-learning surrogate* $f : \mathbf{x} \mapsto P_\Sigma$ where $\mathbf{x} \in R^{2N}$ contains the (x_i, y_i) positions. Such a surrogate reframes wave-farm design as a statistical regression problem, drastically reducing evaluation time while retaining the essential nonlinear mapping learned from high-fidelity simulations.

2. From Physics to Supervised Learning

2.1 Basics of Supervised Learning

The problem we introduced correponds to a supervised machine learning problem. In supervised learning we observe a training set $\mathcal{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^m$ where $\mathbf{x}_j \in R^d$ is a vector of *features* and $y_j \in R$ the corresponding *target*. We aim to find a function f such that

$$y \;=\; f(\mathbf{x}) + \varepsilon, \qquad \text{with} \quad \varepsilon \sim \mathcal{N}(0, \sigma^2),$$

by minimising a loss, typically the mean-squared error MSE $= \frac{1}{m} \sum_j (y_j - f(\mathbf{x}_j))^2$. Since y is here *quantitative*, the task is a **regression** modeling problem.

2.2 Mapping the Wave-farm Problem

Features $\mathbf{x}$ Cartesian coordinates $(X_1, Y_1, \ldots, X_N, Y_N)$ of the N Wave Energy Converters (WECs); hence $d = 2N$ ($d = 98$ for $N = 49$, $d = 200$ for $N = 100$).

Target y Annual average total power P_Σ (column `Total_Power`), computed by the high-fidelity hydrodynamic model

The statistical model f therefore *surrogates* the costly physics-based chain $\big(\mathbf{x} \mapsto P_\Sigma\big)$, enabling instant evaluations for optimisation, uncertainty propagation or real-time control.

3. Dataset Description

3.1 Origin and Size

File	**#layouts** m	**#columns**
`WEC_Perth_49.csv`	36 043	149
`WEC_Sydney_49.csv`	17 964	149
`WEC_Perth_100.csv`	7 277	302
`WEC_Sydney_100.csv`	2 318	302
Total	63 602	—

Each row is one candidate farm layout produced by an evolutionary optimiser; the power values serve as labels for training the surrogate.

3.2 Explanatory Variables

- Xi () – East–West position of buoy i
- Yi () – North–South position of buoy i
- (Optional) a categorical feature Site$\in$ {Perth, Sydney} can be added when concatenating the four files.

3.3 Target Variable

Total_Power () – simulated annual mean power P_Σ of the whole array. Other columns (Power1–PowerN, qW) are themselves outputs of the simulator and should *not* be used as inputs unless a multi-target regression is explicitly desired.

3.4 Data Exploration

Before modeling, we conducted a thorough exploratory data analysis (EDA) to better understand the structure of the dataset and uncover key patterns. We began by examining the distribution of the target variable, Total_Power, for each location. Histograms and kernel density plots revealed that Sydney tends to produce slightly more power on average than Perth, though both cities exhibit relatively similar spread and range.

Boxplots showed slight differences in the variability of, Total_Power, between locations, suggesting possible local environmental influences on wave energy generation. We also computed summary statistics (mean, median, standard deviation, minimum, and maximum) per location, confirming that while Perth and Sydney share similar characteristics, Sydney's energy production is more concentrated around a higher central value.

These visualizations and statistical summaries not only offered insights into the nature of the data but also helped detect potential outliers, confirm data quality, and justify preprocessing choices such as normalization and feature scaling.

We now possess a clean supervised-learning formulation with clearly defined inputs and target, ready for feature engineering and model selection.

4. Methods Description

In this project, we chose to compare two different approaches to modeling the, Total_Power, variable: a linear model and a neural network. This comparison allowed us to evaluate the trade-off between simplicity and interpretability on one hand, and flexibility and predictive power on the other.

4.1 Neural Network Model

We implemented a deep neural network using TensorFlow/Keras. This model is capable of capturing complex nonlinear relationships between the input features and the target variable, which can be especially relevant in the context of physical phenomena such as wave energy generation.

We scaled the data because it helps the model to be more precise. Neural networks rely on gradient-based optimization, which can become unstable when input data are not scaled properly.

We initially built a neural network model to perform the regression and predict the target variable. The architecture consisted of two hidden layers with 64 and 32 neurons, respectively, and the activation function used was the Rectified Linear Unit (ReLU). This model yielded good results, with a mean squared error (MSE) on the order of 10^{-5} and a coefficient of determination of $R^2 = 0.9999$.

We then used this neural network with k-fold cross-validation to strengthen our validation process. The average MSE obtained across the folds was consistent with the previous result, confirming the model's robustness.

For the case involving 100 converters, we employed a slightly larger neural network architecture: two hidden layers with 256 and 128 neurons, respectively. The performance was not as strong as in the previous setup, but still acceptable. The average mean squared error across the 10 folds was:

$$\text{Average MSE over 10 folds:} \quad 0.01$$

4.2 Linear Methods

Linear regression is a natural baseline for regression tasks. It assumes a linear relationship between the predictors and the target variable. Its main advantages are speed, simplicity, and interpretability, especially when the underlying data relationships are close to linear.
To enhance the linear model's performance in the presence of many variables, we also implemented two regularized versions:
Ridge regression (L2 penalty), which helps reduce variance by shrinking the coefficients;
Lasso regression (L1 penalty), which also performs automatic variable selection by driving some coefficients to zero.
All linear models were evaluated using the Mean Squared Error (MSE) on a hold-out test set. Standardization of features was applied beforehand to ensure all variables were on the same scale, which is crucial for regularized models.

We decided, on the other hand, to build a linear regression from the features in order to find a relation with the target variable: **Total_Power**.

This basic linear regression actually works well, with the following metrics output:

$$\begin{aligned} \text{MSE} &= 0.000005 \\ \text{RMSE} &= 0.0023 \\ R^2 &= 1.0000 \end{aligned}$$

We then compared this regression with **Lasso** and **Ridge** regularization.
As with the classical linear regression, Ridge works very well. We obtained the following result:

$$\text{MSE}_{\text{Ridge}} = 0.000005$$

Nevertheless, Lasso is less accurate than both linear and Ridge regressions. We chose a small alpha parameter ($\alpha = 0.001$) in Lasso to avoid a large mean squared error. The result obtained was:

$$\text{MSE}_{\text{Lasso}} = 0.0004$$

We can assume that the lasso does not work as well as the other one because of correlation between variable. Also it may be because of each variable is important.

5. Conclusion

In this study, we evaluated and compared two main predictive approaches for modeling wave energy output (`Total_Power`) from physical and environmental features: **linear models** (including Ridge and Lasso regularization) and **neural networks**.

Both the *basic linear regression* and the *neural network (MLP)* produced excellent results. Using the **Mean Squared Error (MSE)** and the $\mathbf{R^2}$ coefficient of determination as comparison metrics:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad \text{and} \quad R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

we found that both models achieved **near-perfect fits**, with MSEs as low as `0.000005` and R^2 values equal to `1.0000`. This strongly suggests that the relationship between the features and the target variable is highly predictable using both approaches.

However, while the **neural network** is capable of capturing non-linear patterns and complex interactions, we observed that it did *not outperform* the simpler linear model in this particular case. In fact, the **linear regression performed just as well**, even without regularization.

From a practical perspective, the linear model is significantly less computationally expensive. It requires almost no hyperparameter tuning, trains in a fraction of the time, and remains fully interpretable which is a key advantage in applications where understanding the model's decisions is important. The neural network, while effective, introduces higher training time and greater complexity.

Therefore, we recommend the linear regression (or Ridge regression or Lasso regression) as the preferred model for this task. It offers equivalent performance to the neural network but with lower computational cost and better interpretability.

As for data processing, our results suggest that the existing features are well structured and informative, and the preprocessing pipeline (notably standardization) was sufficient to ensure model performance. A possible area for improvement could involve analyzing feature importance or redundancy more deeply, or testing hybrid models that combine interpretability and flexibility.