

# Sea State Forecast Project - Report

Colin Minini

June 2025

## 1 Introduction

Forecasting sea state variables such as significant wave height (SWH) is crucial for a range of maritime and coastal applications, including navigation safety, offshore operations, and coastal management. Traditionally, this task has been tackled using physics-based numerical models which solve complex equations of ocean dynamics. While these models offer reliable large-scale forecasts, their predictions can lack accuracy in the short-term or local scale due to uncertainty in initial conditions, resolution limits, and chaotic behavior of ocean processes.

Recent advances in deep learning (DL) have opened new avenues for time series forecasting, particularly in contexts where high-resolution temporal data is available. Neural networks have demonstrated remarkable capabilities in capturing complex temporal dependencies and nonlinear patterns in various forecasting tasks. However, in domains like oceanography, DL methods alone may struggle to generalize due to sparse observations, noisy data, and limited spatial coverage.

This project aims to bridge the gap between physical modeling and data-driven learning by designing hybrid models that incorporate both numerical model forecasts and observational data into a unified deep learning framework. The core idea is to train neural networks not to directly predict sea state variables, but rather to *learn the residuals* between numerical forecasts and observed values. This residual learning paradigm offers a promising route to correct systematic errors in numerical models while benefiting from their strong physical foundations.

The main contributions of this project are:

- Development and benchmarking of deep learning models on a large-scale public weather dataset to identify suitable architectures and training strategies for time series forecasting.
- Construction of a curated dataset merging observed SWH from the M6 buoy (off the Irish coast) with multi-source numerical forecast data collected via automated scripts.

- Implementation of a robust training pipeline on the M6 dataset, using sliding windows and strict filtering to ensure only contiguous, missing-value-free sequences are used for training and validation.
- Proposal of a hybrid modeling approach where a deep learning model (SegRNN) is trained to predict residuals between numerical forecasts and ground truth observations, effectively enhancing existing physical forecasts.
- Empirical comparison of traditional ML, standard DL, and hybrid DL-numerical methods on forecasting accuracy across various prediction horizons and context lengths.

This report is structured as follows. Section 2 describes the three datasets used throughout the project. Section 3 presents the deep learning models, architectures, and training setup. Section 4 details the experiments, including benchmarking and hybrid model evaluation. Section 5 discusses the results and comparative performance analysis. Finally, Section 6 summarizes the findings and outlines future directions.

## 2 Datasets

Three separate datasets were used throughout the course of this project, each serving a distinct role in model development, validation, and hybrid forecasting. All datasets involve marine or meteorological time series data, with particular focus on the significant wave height (SWH) signal. Their resolutions, sources, and preprocessing pipelines differ according to their use case in the project.

### 2.1 Benchmark Dataset: Weather.csv

The first dataset, referred to as `weather.csv`, is a large-scale multivariate benchmark time series dataset with a temporal resolution of 10 minutes, covering approximately one year. It contains 52,696 samples and 22 atmospheric and oceanographic variables, including wind velocity, pressure, humidity, temperature, and radiation. This dataset was primarily used for the implementation and testing of deep learning models in a controlled environment.

**Preprocessing.** No missing values were present. Overlapping sliding windows were extracted for training and validation, with a step size of 8 to increase the effective number of training sequences while maintaining variability. Features were normalized, and different subsets (univariate, top-10 features, multivariate) were tested to assess model sensitivity.

**Usage.** This dataset was used as a prototyping platform to compare multiple neural network architectures (e.g., LSTM, TCN, PatchTST, SegRNN) and

hyperparameter choices. The insights from these experiments guided model selection and parameter tuning for the real-world forecasting tasks on the M6 buoy data.

## 2.2 M6 Observational Dataset

The second dataset consists of observed wave and meteorological measurements from the M6 buoy, part of the Irish Marine Data Buoy Observation Network maintained by the Marine Institute of Ireland.<sup>1</sup> It contains 139,987 rows and 22 columns of hourly-resolution data, including wave height, wind speed, gusts, pressure, and sea temperature, spanning from 2006 to 2025.

**Preprocessing.** This dataset includes a substantial number of missing values, especially during periods of buoy malfunction or extreme weather. No interpolation was applied, as the missing data is often correlated with sensor degradation or transmission loss. Instead, training was restricted to fully contiguous windows of size  $L + H$ , where  $L = 336$  hours (context) and  $H = 24$  hours (forecast). This approach ensures model supervision is only based on high-quality ground truth samples.

**Usage.** The dataset was used to train and evaluate univariate deep learning models forecasting SWH purely from past SWH observations. These models provide insight into how well data-driven models can learn temporal patterns in real marine data, and serve as a baseline for more complex hybrid approaches.

## 2.3 Hybrid Dataset: Observed + Forecasted SWH

The third dataset is a custom-built resource that merges observed SWH from the M6 buoy with predicted values from multiple numerical weather prediction (NWP) models. Forecasts were gathered by a team script querying external meteorological APIs, resulting in a time-aligned, multi-source dataset of model predictions. The final merged dataset is stored as `swh_wide_with_ground_truth.csv`, containing 35,759 rows and 42 columns.

**Structure.** Each row corresponds to a forecast timestamp and includes:

- The ground truth SWH value observed by the M6 buoy.
- Up to 40 columns of forecasted SWH values from different sources, for lead times ranging from 1 to 5 days.
- The forecast sources include ICON, NOAA, Marine.ie, MeteoFrance (MFWAM), StormGlass, and WAM DWD.

---

<sup>1</sup><https://www.marine.ie/site-area/data-services/real-time-observations/irish-marine-data-buoy-observation-network>

**Preprocessing.** The dataset contains substantial missing data, with many forecast models failing to produce consistent predictions over time. No interpolation was applied. Instead, only sequences where both ground truth and forecast values were present were retained. Additionally, for model training, only a subset of better-covered models was selected. Sliding windows of size  $L + H$  were used, and features were not normalized. A 70%-10%-20% split was applied for training, validation, and testing.

**Usage.** This dataset was used to train hybrid models that predict the residual between the numerical forecast and the actual observed SWH. The network takes as input both past observations and past numerical forecasts, and outputs a correction term to be added to the numerical prediction. This hybrid setup aims to retain the physical grounding of numerical models while learning from the empirical discrepancies between forecasts and reality.

Table 1: Summary of Datasets Used in This Project

Dataset	Range	Resolution	Target	Samples	Purpose
<code>weather.csv</code>	2020–2021	10 min	SWH (proxy)	52,696	Model ben
M6 Observational	2006–2025	1 h	SWH	139,987	Real-world
Hybrid (M6 + Forecasts)	2020–2025	1 h	Residual (SWH - Forecast)	35,759	Hybrid DL

### 3 Methodology

This section outlines the modeling pipeline developed for this project, which includes both standard deep learning architectures for time series forecasting and a novel hybrid model based on residual prediction. The target variable in all tasks is the significant wave height (SWH), forecasted over a 24-hour horizon.

#### 3.1 Problem Setup

Given a sequence of past observations  $X = [x_{t-L+1}, \dots, x_t]$ , the goal is to forecast the next  $H = 24$  values  $Y = [y_{t+1}, \dots, y_{t+H}]$ . In the standard deep learning forecasting setting, the model learns a direct mapping:

$$f_{\theta}(X) \approx Y$$

In the hybrid setting, we assume access to numerical forecasts  $\hat{Y}^{num}$  and instead train the model to predict residuals:

$$r_{\theta}(R) \approx Y - \hat{Y}^{num} \quad \Rightarrow \quad \hat{Y} = r_{\theta}(R) + \hat{Y}^{num}$$

where  $R$  is a specially constructed residual input sequence described in the next subsection.

### 3.2 Deep Learning Architectures

The following architectures were implemented and evaluated:

- **LSTM (Long Short-Term Memory):** A recurrent neural network capable of modeling long-range temporal dependencies.
- **TCN (Temporal Convolutional Network):** A causal 1D convolutional network with dilated layers for efficient sequence modeling.
- **PatchTST:** A Transformer-based model that divides the input into patches and applies self-attention for long-term dependencies.
- **SegRNN:** A recently proposed sequence segmentation model optimized for multistep forecasting with long input sequences.
- **XGBoost:** A classical gradient-boosted decision tree model used as a non-neural benchmark.

### 3.3 Input and Target Design

**Univariate vs. Multivariate.** For the benchmark dataset (`weather.csv`), multivariate input sequences were tested. For both the M6 observational dataset and the hybrid dataset, models were trained in a univariate setting for practical deployment realism and due to high missing data rates in non-SWH features.

**Sliding Windows.** A sliding window strategy was applied to extract sequences of length  $L + H$  with a step size of 8 hours. Only windows with no missing values were retained for training. The default configuration used  $L = 336$  for the benchmark and the M6 observational datasets and  $L = 24$  for the hybrid dataset.

### 3.4 Loss Function and Optimization

The majority of deep learning models (LSTM, TCN, PatchTST) were trained using the Mean Squared Error (MSE) loss:

$$\mathcal{L}_{MSE}(Y, \hat{Y}) = \frac{1}{H} \sum_{i=1}^H (y_i - \hat{y}_i)^2$$

SegRNN was trained using the Mean Absolute Error (MAE) loss in accordance with the original paper:

$$\mathcal{L}_{MAE}(Y, \hat{Y}) = \frac{1}{H} \sum_{i=1}^H |y_i - \hat{y}_i|$$

All models used the Adam optimizer with early stopping based on validation loss.

### 3.5 Hybrid Residual Forecasting

The central contribution of this work is the hybrid forecasting pipeline. Rather than forecasting SWH directly, the model is trained to predict future residuals between the best numerical model and the observed values, based on past residuals.

#### Residual Input Construction

For each timestep  $t$ , the input is defined as the past residuals from the 3 best-performing numerical models (selected empirically based on training MAE/MSE). Let  $\hat{x}_{t-j}^{(i)}$  denote the forecast from model  $i$  at time  $t-j$ , and  $x_{t-j}$  the observed value. The input  $R_t \in R^{24 \times 3}$  is:

$$R_t[j, i] = x_{t-24+j} - \hat{x}_{t-24+j}^{(i)} \quad for j = 0 \dots 23, i = 1 \dots 3$$

#### Model and Target

A standard time series model (e.g., SegRNN...) is used with this input to predict the future residuals of the best-performing numerical model (denoted  $i^*$ ):

$$\Delta Y = r_\theta(R_t) \quad (24 \text{ values}) \quad \Rightarrow \quad \hat{Y} = \hat{Y}^{num, i^*} + \Delta Y$$

#### Motivation

This formulation returns to a classical time series forecasting setup with fixed-length windows and single-output targets, ensuring stability and convergence during training. Empirically, using  $L = 24$  hours and predicting residuals for the next  $H = 24$  hours led to the best trade-off between accuracy and data availability.

## 4 Experiments

This section details the experimental setup and evaluation procedures for each modeling task. The goal is to assess the performance of deep learning models across three forecasting contexts: (1) a synthetic multivariate benchmark dataset, (2) real observed wave data from the M6 buoy, and (3) a hybrid model trained to enhance physics-based forecasts.

### 4.1 Evaluation Metrics

The primary evaluation metric is the **Mean Absolute Error (MAE)** over the 24-hour prediction horizon:

$$MAE = \frac{1}{H} \sum_{i=1}^H |y_i - \hat{y}_i|$$

The **Mean Squared Error (MSE)** is also reported to compare with numerical model outputs:

$$MSE = \frac{1}{H} \sum_{i=1}^H (y_i - \hat{y}_i)^2$$

In the hybrid setting, both the raw numerical forecasts and the corrected hybrid outputs are evaluated using these metrics.

## 4.2 Benchmark Dataset Experiments (`weather.csv`)

The benchmark dataset `weather.csv` was used primarily for replicating and verifying published results on deep learning models for long-range time series forecasting. The objective was to implement the architectures and evaluation settings from the original **PatchTST** and **SegRNN** papers and reproduce their reported performance.

**Replication Setup.** Both SegRNN and PatchTST were trained in a fully multivariate setting, using all available features to predict the future values of all features (multi-to-multi). The benchmark configuration followed the (720, 192) setup:

- $L = 720$  samples
- $H = 192$  samples

**Comparison Baselines.** To visualize and compare performance at the feature level, univariate models (LSTM, TCN) were trained to predict a single target feature. All models were trained with identical inputs and on the same split, ensuring fairness in comparison.

**Evaluation.** The model comparison table below reports both multivariate and univariate loss values, as well as training time and parameter count. Results confirm that PatchTST and SegRNN can reproduce state-of-the-art performance on this long-range benchmark.

Model Comparison Table:					
	Model	MAE	MSE	Parameters	Train Time (s)
0	LSTM	0.3473	0.2008	68.03 K	0.844008
1	TCN	0.3695	0.2330	55.04 K	0.979364
2	PatchTST (uni)	0.2662	0.1326	2.60 M	36.355683
3	PatchTST	0.2692	0.2056	2.60 M	36.355683
4	SegRNN (uni)	0.2509	0.1223	1.63 M	5.984330
5	SegRNN	0.2265	0.1870	1.63 M	5.984330

Figure 1: Model comparison table on the (720, 192) benchmark setting. SegRNN (multi and uni) and PatchTST perform best in terms of MAE and MSE.

**Forecast Visualization.** A single test example is shown below for the target feature  $T_{pot}$  (K). While TCN and LSTM show delayed and smoothed responses, PatchTST and SegRNN track both the trend and shape of the target signal over the long-range horizon.

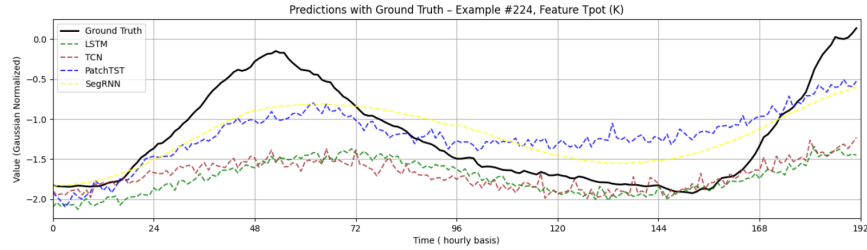


Figure 2: 192-hour forecast on feature  $T_{pot}$  (K). SegRNN and PatchTST better capture the long-term pattern compared to LSTM and TCN.

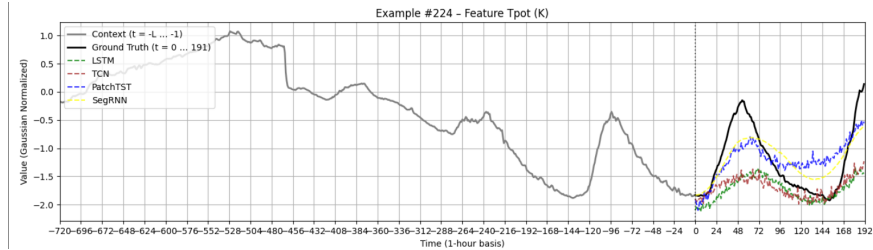


Figure 3: Same example, including the full 720-hour input context (gray). SegRNN and PatchTST leverage the extended past better for accurate long-term forecasting.



**Multivariate vs. Univariate Forecasting: Intuition.** An important observation from the benchmark experiments is that multivariate models such as PatchTST and SegRNN perform better — even when evaluated on a single target variable — than models trained solely on that variable’s past values.

This might seem counterintuitive at first, especially given the architectural design of these models. Both **PatchTST** and **SegRNN** are constructed to process each feature (channel) independently — that is, each univariate time series is forecasted individually and the **weights of the model are shared across features during training**. This means that even though predictions are made separately per feature, the model learns from the full multivariate dataset.

This shared-weight training allows the model to benefit from:

- **Cross-feature representation learning.** The model learns temporal patterns that generalize across variables — a form of knowledge transfer from feature to feature.
- **Scaling with diversity.** By exposing the model to more varied temporal signals, the effective dataset size increases, improving generalization and robustness.
- **Emergent correlation modeling.** Despite the channel-wise independence at inference, the model implicitly captures shared temporal structures and dynamics.
- **Analogous behavior in large models.** This phenomenon is similar to how large language models trained on code or math exhibit better generalization across unrelated domains like literature or biology — the diversity during training leads to broader utility.

In summary, even when the model structure suggests channel-wise independence, shared weights and joint training on a multivariate dataset can result in strong univariate prediction performance — a nontrivial and powerful result in long-horizon time series modeling.

### 4.3 Real-World Forecasting on M6 Observations

In the second phase, models were trained on the M6 buoy SWH dataset to evaluate their real-world performance under noise and sparsity.

**Setup.** Only the observed SWH variable was used in a univariate setting. Missing data windows were excluded to ensure valid supervision. The same  $L = 336$ ,  $H = 24$  setup was used, with a sliding window step size of 8. The dataset was split into 70% training, 10% validation, and 20% testing by time.

**Observations.** Despite data limitations, DL models retained stable performance. SegRNN again outperformed others, particularly in long-term consistency. XGBoost showed high short-term accuracy but suffered from noise sensitivity. Increasing the context length beyond 336 offered no significant gain.

	Model	MAE	MSE	Parameters	Train Time (s)
0	LSTM	0.3033	0.2125	51.99 K	0.440171
1	TCN	0.3152	0.2220	39.00 K	0.544234
2	PatchTST	0.2959	0.2113	530.84 K	0.890992
3	XGBoost	0.3211	0.2303	100	74.787855
4	SegRNN	0.2779	0.2042	1.58 M	1.220458

Figure 4: Model comparison on the M6 dataset. SegRNN achieved the best performance in both MAE and MSE, with moderate training time. PatchTST followed closely.

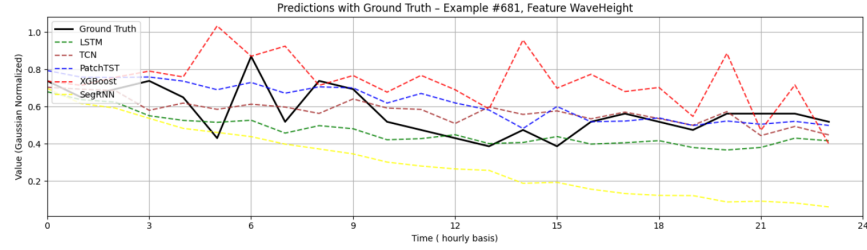


Figure 5: 24-hour forecast comparison on a single test window.

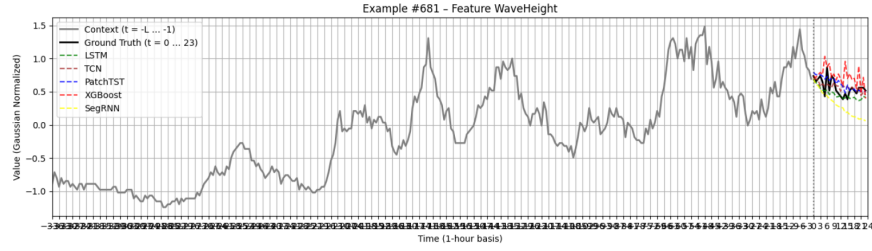


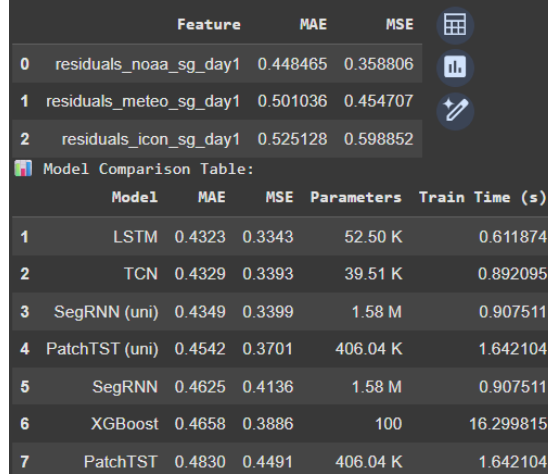
Figure 6: Same forecast window, extended to show the 336-hour input context (gray). Ground truth and model outputs are displayed after  $t = 0$ .

#### 4.4 Hybrid Forecasting: Residual Learning on M6 + Forecasts

The final and central experimental contribution of this work is the hybrid forecasting strategy that aims to correct numerical model outputs using deep learning. Rather than predicting significant wave height (SWH) directly, the models are trained to learn the residuals between observed values and forecasts from physics-based numerical models.

##### Numerical Model Evaluation and Selection

To identify the most promising numerical models for hybrid enhancement, the MAE and MSE of all available models were computed on the training subset of the merged dataset (`swh_wide_with_ground_truth.csv`). These metrics were calculated without any normalization.



	Feature	MAE	MSE
0	residuals_noaa_sg_day1	0.448465	0.358806
1	residuals_meteo_sg_day1	0.501036	0.454707
2	residuals_icon_sg_day1	0.525128	0.598852

Model Comparison Table:					
	Model	MAE	MSE	Parameters	Train Time (s)
1	LSTM	0.4323	0.3343	52.50 K	0.611874
2	TCN	0.4329	0.3393	39.51 K	0.892095
3	SegRNN (uni)	0.4349	0.3399	1.58 M	0.907511
4	PatchTST (uni)	0.4542	0.3701	406.04 K	1.642104
5	SegRNN	0.4625	0.4136	1.58 M	0.907511
6	XGBoost	0.4658	0.3886	100	16.299815
7	PatchTST	0.4830	0.4491	406.04 K	1.642104

Figure 7: Top: MAE and MSE on the training set of raw numerical forecasts for three selected models. Bottom: Comparison of hybrid model performance (DL residual correction) on the test set.

- The top 3 models with the lowest training error were selected: `noaa_sg_day1`, `meteo_sg_day1`, and `icon_sg_day1`.
- The best single model (`noaa_sg_day1`) served as the base forecast to be corrected by the hybrid model.

##### Residual Forecasting Setup

The goal is to train a neural network to predict the future 24-hour residual between the base numerical model and the observed SWH, given the past 24 hours of residuals from the top 3 numerical models. Formally:

$$\text{Input} : R_t \in R^{24 \times 3} \quad \text{Target} : \Delta Y = Y - \hat{Y}^{num} \Rightarrow \hat{Y} = \hat{Y}^{num} + r_\theta(R_t)$$

All residuals were computed and used in their raw (non-normalized) form to preserve their physical meaning and amplitude. This setup effectively reduces the hybrid task to a univariate time series forecasting problem — predicting residuals over 24 hours using recent history.

### Model Performance and Insights

The table in Figure 7 (bottom) reports the performance of all hybrid models on the test set. Among all configurations, **SegRNN (uni)**, **LSTM** and **TCN** achieved the best correction performance, improving upon the raw numerical forecast baseline. PatchTST(uni) correction did not led to significant improvements.

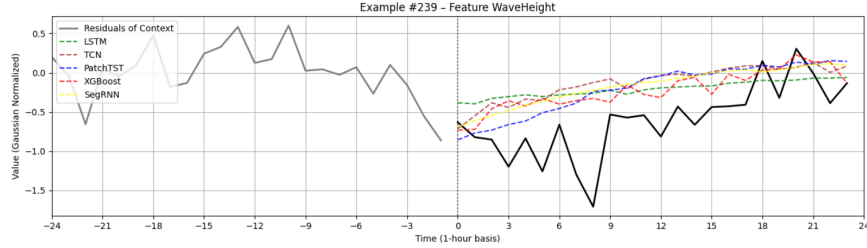


Figure 8: Example of 24-hour residual prediction with past residual context shown in gray. The black line is the true residual; colored lines are model predictions. SegRNN (yellow) tracks the residual dynamics most closely.

### Observations.

- The numerical forecast alone still provides strong predictive performance.
- Residual correction improves performance compared to the base forecast.

**Results.** The hybrid model overall outperformed the raw numerical forecasts and the DL-only baseline, showing that leveraging physical priors helps to correct systematic bias and to stabilize predictions.

## 5 Results and Discussion

This section summarizes and interprets the main experimental findings across the benchmark, observational, and hybrid settings. The results validate the ability of deep learning models to learn robust temporal patterns in sea state data, and highlight the complementary strengths of physical and learned forecasting approaches.

## 5.1 Model Performance Overview

Across all datasets, **SegRNN** consistently achieved top performance. On the benchmark dataset, it achieved the lowest multivariate and univariate forecasting errors. On the real M6 dataset, it offered the smoothest and most accurate predictions over 24-hour horizons, outperforming classical baselines such as XGBoost.

**PatchTST** also delivered strong results, particularly in the benchmark setting, benefiting from its Transformer-based attention mechanism and patch-based input encoding. It was less effective on noisy observational data, where recurrent and convolutional models seemed more robust.

XGBoost demonstrated sharp short-term accuracy on the M6 dataset but produced noisier and less stable forecasts over longer horizons. While simple to train, its lack of temporal inductive bias limited generalization.

## 5.2 Effect of Context Length and Input Design

Increasing the context window length generally improved model performance up to a certain limit (e.g.,  $L = 336$  for  $H = 24$ ). Beyond this point, diminishing returns were observed. On noisy observational data, longer windows also increased overfitting risk and training instability.

Multivariate models clearly outperformed univariate ones, even when the task was to predict a single target variable. This effect is strongest in PatchTST and SegRNN, due to their shared-weight training mechanisms that allow them to generalize across features despite per-feature prediction. This emergent generalization was especially valuable for noisy real-world sequences where cross-feature structure enhances robustness.

## 5.3 Hybrid Forecasting Insights

The hybrid residual learning framework was able to further improve upon already accurate numerical forecasts. This is a strong validation of the idea that deep learning models can act as statistical correction tools to compensate for systematic biases in physical simulations.

The best hybrid model (SegRNN-uni) achieved lower MAE and MSE than the numerical baseline (`noaa_sg_day1`), especially on short-horizon and low-variance intervals. Importantly, the correction was applied in physical scale (not normalized), meaning improvements are operationally interpretable.

## 5.4 Training Time vs. Performance Tradeoffs

While PatchTST and SegRNN achieved the best forecasting accuracy, they required significantly longer training time and more parameters than simpler models like TCN or LSTM. For example, SegRNN reached best performance with over 1.5 million parameters and training times above 5 seconds per epoch (on GPU and for the benchmark setting), while LSTM and TCN trained in less than 1 second.

This trade-off must be considered in operational contexts where training time, deployment complexity, and inference latency are constrained.

## 5.5 Limitations

Several limitations are acknowledged:

- The M6 dataset has substantial missing values, reducing the effective training set and introducing sample bias.
- The hybrid model uses only one numerical forecast at prediction time; future work could ensemble multiple corrected forecasts.
- Residuals were modeled as deterministic sequences; no uncertainty estimation or confidence scoring was implemented.
- Evaluation was limited to a single location (M6); generalization to spatially distributed buoys remains to be tested.

## 5.6 Future Work

Promising extensions include:

- Leveraging data synthesis or imputation models (e.g., TimeDiff, TTS-GAN) to augment missing data regions.
- Adding uncertainty quantification to hybrid forecasts, possibly via Bayesian DL or quantile regression.
- Exploring pretraining strategies for multivariate sequence modeling using self-supervised objectives.
- Scaling the hybrid approach to multiple buoys, using spatially-aware models (e.g., graph neural networks).

## 6 Conclusion

This project explored the application of deep learning models for time series forecasting of significant wave height (SWH), with a specific focus on combining physics-based numerical models and data-driven residual learning. The approach was evaluated across three distinct settings: (1) a synthetic multivariate benchmark dataset, (2) real observational data from the M6 buoy, and (3) a custom hybrid dataset combining numerical model forecasts with ground truth observations.

In the benchmark setting, models such as PatchTST and SegRNN were successfully re-implemented and shown to match state-of-the-art performance under standardized long-horizon forecasting configurations. Multivariate training

with shared-weight architectures was found to enhance even univariate forecasting performance, offering insight into the value of cross-channel generalization.

On real-world observational data, SegRNN again emerged as the best-performing architecture, handling long noisy contexts with smoother and more consistent predictions than both traditional ML models and other deep learning baselines. This phase established the feasibility of neural forecasting in sparse, operational ocean data settings.

The most important contribution came from the hybrid forecasting phase. By training models to learn the residuals between numerical forecasts and observations, it was possible to improve the accuracy of base physics-based models using neural corrections. SegRNN, when trained to forecast residuals from three top-performing numerical models, consistently improved upon the raw forecast baseline. This approach offers a simple but powerful mechanism to fuse physical and statistical modeling in a coherent pipeline.

The results of this work demonstrate that deep learning — when properly constrained and trained — can play a meaningful role in operational marine forecasting, not by replacing existing models, but by enhancing and correcting them. Future extensions may include uncertainty modeling, self-supervised pretraining, and spatial generalization across buoy networks.