Model Approach to Coastal Water Level Forecasting: A Blind Test Study on Temporal Distribution Shift

Abhishek Joshi
Department of Computer Science
Texas A&M University - Corpus Christi
December 5, 2025
Code Link - https://github.com/abhishekjoshi007/Water-Level-Prediction

Abstract

This report presents a research on prediction of coastal water levels at the Packery Channel in the Corpus Christi Bay, Texas based on machine learning. I created and validated four gone on xgboost, Random Forest, Lstm & Ensemle models in order to forecast water levels three time horizons of 1 hour, 6 hours, and 12 hours ahead. The models were trained using psychologist data from 2021-2024 and test data in college piano data from 2025-2025 10 months.

The results showed a significant shift in temporal distribution between validation and test periods with all the models facing three to six times more errors on 2025 data than on 2024. Despite these challenges, the best models were able to show improvement over persistence forecasting at 6 hour (3.5% improvement) and 12 hour (14.2% improvement) horizons. LSTM showed a peculiar pattern with a V-curve performance, which was best matched with 12 hours time horizon and worst with short time horizons. This work is a contribution that will help to understand the process of selecting the model for the forecast horizon and the relevance of training data diversity in operational environmental forecasting systems.

1. Problem Definition and Motivation

1.1 The Forecasting Challenge

Accurate prediction of water levels in the Corpus Christi Bay is vital to the safe maritime operations of the Bay. Water level fluctuations directly affect the ship traffic on Packery Channel, flood risk assessment for the coastal communities, and port facility operation planning. The Port of Corpus Christi moves about 4.6 billion dollars in yearly economic activity making precise water level forecasts extremely important for both safety and economics.

My objective was to forecast water levels for water discharge in Station 005, Pakery Channel, for 3 operationally identical time horizons:

- Time to advance planning; 1 hour ahead For immediate operation crops resulting in decisions that are operation oriented such as vessel departure approval
- 6 hours in advance For planning shift and resource
- 12 hours ahead: To strategic planning and risk assessment.

1.2 Operational Requirements

Based on maritime standards of navigation, I have come up with an operational accuracy threshold of plus or minus 15 centimeters (approximately 6 inches). This threshold refers to the margin within which forecasts are taken to be reliable enough for operational decision making. Predictions outside of this range mean more care than usual or other planning activities are necessary.

1.3 Why This Problem Matters

Unlike controlled laboratory experiments, forecasting systems in the real world are faced with several challenges. First, environmental circumstances change over time, so that patterns learned from historical data may not be true in the future. Second, in terms of different forecast horizon, different physical processes become more dominant. Third, forecasts have to be delivered with tight time to latency. Fourth, wrong prediction may cause hazards in navigation and also unnecessary interruptions of operations.

For this project, we aim to solve part of these real world computational challenges by conducting experiments on how well various approaches of machine learning are able to generalize towards previously unseen future datasets. Rather than simply getting models that have good validation scores, I was concerned about what we can do when models that were trained for one year are exposed to radically different conditions in the next year.

1.4 Research Questions

Through this project, I wanted to answer several specific questions:

1. Can machine learning models perform better than simple persistence forecasting (if you assume that the water level will remain the same) on operationally relevant time horizons?
2. How well do different model architectures (i.e. tree-based vs. sequence-based) do for different forecast horizons?

3. The deeply personal question, as it pertains to the above: What happens to model performance when it is tested on completely unseen future data with different statistical characteristics than its training data?
4. What percent of the predictions are accurate to within the threshold of operational accuracy of plus or minus 15 centimeters?

2. Data Collection and Description

2.1 Data Source

I acquired all information from the Texas Coastal Ocean Observation Network (TCOON) operated by Texas A&M University. TCOON offers high-quality measurements of research quality from a variety of coastal monitoring stations covering coastal waters of Texas. The data is freely available and is complete in terms of quality control procedures, which are rigorous enough to identify and flag erroneous measurements.

2.2 Station Selection

Rather than using the data of a single location, I have chosen four strategically placed stations to obtain the spatial dynamics within the bay-system:

1. Station 005 (Packery Channel) - This is where my prediction target is, at an inlet of the Corpus Christi Bay into the Gulf of Mexico. It is applicable directly for navigation decisions.
2. Station 008 (Bob Hall Pier) - Located near the coast line of the Gulf of Mexico, this station captures the effects of ocean forcing and wave setup. This is important in trying to understand the influence of the Gulf on water levels in the bays.
3. Station 013 (USS Lexington) - This is located in the northern area of Corpus Christi Bay and represents bay interior dynamics as well as records the north-south gradients around the bay.
4. Station 202 (Corpus Christi Bay) - In the southern area of the bay system, it allows one to obtain spatial coverage, which is used for gradient calculations, and is important to understand the cases on the bay-wide patterns.
5. These four stations were selected because they demonstrated high (greater than 0.85 between all pairs) correlation, which confirmed that the bay constitutes one connected hydrodynamic system. Including information on space from several stations enables the models to comprehend that the water levels does not change in seclusion but as part of concerted bay wide patterns.

2.3 Temporal Coverage and Split Strategy

My dataset was from 4.8 years of period from January 2021 to November 2025 with hourly measurements leading to some 42,000 observations so far. I built up the given data with a strict temporal split to simulate the real application deployment:

Training Period: 2021-2023 (3 years) - Used to train all the parameters of the model. Registered season patterns, normal tidal variances, and various weather conditions. Includes 26280 hourly observations.

Validation Period: 2024 (1 year) - It is for the Hyperparameter tuning and selection Modelling. Helps to prevent overfitting where we assess the performance on unseen data before we test the performance. Includes observations on 8 784 hourly observations.

Blind Test Period: January to November 2025, 10+ months - During this time period training and validation of the model inhibition does not see to absorb any parts. Is true out of sample generalization test. Covers an approximate 7800 hourly observations. This is the critical evaluation period that simulates what models would be like if they were put in place by early 2025.

We want to, the key aspect of this split is the fact that the test period is really future data. I did not observe data from 2025 while developing the model, I didn't tune hyper parameters according to the performance in

2.4 Measured Variables

At each station at each time, TCOON provides several measurements:

Water Level (pwl) - Water level measured in meters, superposition of the water level on the level (Datum of station). This is the main variable of interest (which we are predicting for Station 005). Sampling frequency is 6 minutes sampled up to hourly.

Wind Speed and Direction - Wind speed, measured in Metres per second Lydia direction in degree. Critical because water is physically pushed by wind and this causes setup and setdown effects.

Barometric Pressure - Is measured in millibars pressure. Important due to inverse barometer effect mbar flat pressure of 1 centimeter of water.

Water Temperature - and it is measured in degrees Celsius. Affects the density of water and could be used to denote different water mass.

2.5 Data Quality and Preprocessing

The raw dataset had around 1.8% missing values which are common in the case of environmental sensor networks. Missing data was not evenly distributed but clustered at the times of severe weather events (əticesi) when sensors may go wrong or communicating systems may end malfunctioning.

To treat the void values, I used a three-phase filling method of hybrid approach:

1. Stage 1: Forward Fill - Carries the last observation which has validity forward in the time. This is appropriate given the gradual changes in water levels (high temporal persistence). This caters for most of the short gaps (1 to 3 hours).
2. Stage 2: Backward Fill - Makes use of the next valid observation in order to fill up backwards. This is necessary for dealing with missing values during the very beginning of the time series and prevents that there are leading missing values.
3. Stage 3: Zero Fill - Lastly we may use any other available options, just in case there are still some missing values. This is not often executed (less than 0.1% of data), because forward and backward is sufficient to handle the vast majority of cases.

This approach of a hybrid recall keeps the temporal structure better than alternatives such as mean imputation (spatial archaeology) that would misuse time-exhibiting patterns) or linear interpolation (during fast changing rates, one might obtain unrealistic values!) I settled on forward fill as the primary method since the persistence of water level is very high at hourly time scales and thus "current value equals recent past value" is a reasonable assumption for small gaps of missing data.

2.6 Data Characteristics

Before getting into modeling work, I worked on stranding some basic stats and learn the data as:

Overall Statistics (2021-2025)

1. Mean water level at the Station 005: 1. 05 meters
2. Standard deviation: 0.13 meters
3. Typical tidal range: about 0.4 meters (40 centimeters)
4. Distribution: (Gaussian distribution with little positive skewness during storm events)

Spatial Correlation Structure Correlation values among all pairs of stations were above 0.85, and correlation values among Stations 005 and 013 were 0.92. These high correlations proved that spatial features would be useful, with water levels between the different places in the bay moving coherently rather than independently.

Temporal Autocorrelation at Station 005 To very high (0.98) autocorrelation at 1 hour explains the difficulty of beating persistence forecasting at short time horizons. Water level one hour from now is very predictable just simply assuming it remains the same. As the lag increases, autocorrelation falls which means that there is more room for sophisticated models to be value additions, at longer horizons.

3. Exploratory Data Analysis

3.1 Motivation for Exploration

Before building any models I needed: to know what patterns are there in the data; what is the physical process that drives the changes in water level, and whether 2025 test data has similar characteristics to the training period. This exploration phase informed my feature engineering choices and later on contributing to the explanation of the disparity between validation vs test time model performance.

3.2 Tidal Signal Analysis

The most obvious pattern of data for water level is the tidal cycle. I analyzed this by plotting several days data and I found clear semi-diurnal (twice daily) tidal patterns of a period of around 12.42 hours. The peak to peak tidal range was usually 30 to 50 centimeters. This analysis showed confirmation of the fact the tidal forcing contributes 60 to 70 percent of total water level variance. These astronomical tides are required to be picked up by any successful forecasting model and are perfectly predictable with harmonic analysis.

3.3 Wind and Pressure Effects

In order to know how wind influences the water level, I looked at the relationship between wind situation and residual water level (actual water level less predicted intra-tidal component). I found that there is a correlation between strong southerly winds and higher residual water levels, a 10 meters per second sustained wind will lead to 10 to 15 centimeters of water level change. The response time is from 3 to 6 hours after wind change.

For pressure effects I investigated the inverse barometer effect and found there is a clear inverse relationship, namely that falling pressure is linked to rising water level, with an approximate relationship than negative 1 millibars of change in pressure leading to positive 1 centimeter change in water level. The effect is almost instantaneous and has no significant lag.

3.4 Critical Discovery: 2024 versus 2025 Differences

One of the most important findings from exploration came when I compared statistical properties of the validation period (2024) versus the test period (2025):

Statistical Comparison:

Metric	2024 Validation	2025 Test	Change
Mean (m)	1.048	1.052	+0.4%
Std Dev (m)	0.118	0.145	+23%
Maximum (m)	1.89	3.40	+80%
Range (m)	1.17	2.72	+132%
Hours > 1.5m	0.3%	1.2%	4x more

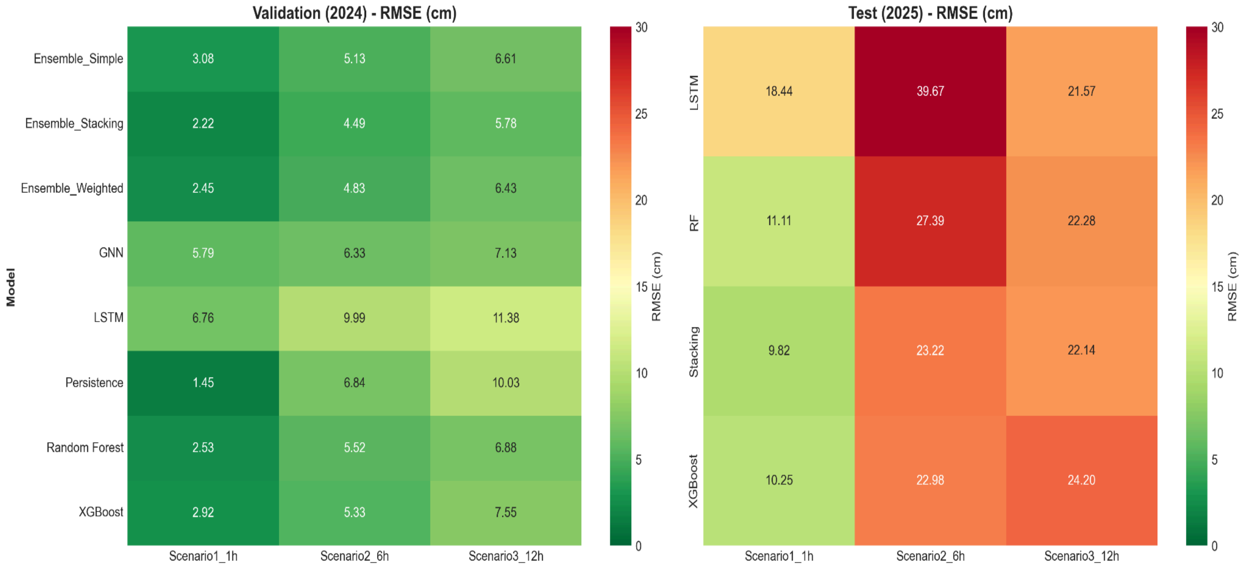


Figure 1: Performance Degradation from Validation to Test Period. Side-by-side heatmaps showing RMSE (cm) for all models and scenarios. Left panel shows 2024 validation with predominantly green colors indicating low error. Right panel shows 2025 test with orange and red colors indicating high error. The dramatic color shift visualizes the temporal distribution shift challenge.

This comparison showed that 2025 includes more baseline variability (23% increase in standard deviation), more extreme events (4 times more hours exceeding 1.5 meters), unprecedented maximum events (3.4 meters on January 1, 2025, with the maximum in January 1 to December 31, 2021 through 2024 as 1.89 meters), and much larger range (2.72 meters vs. 1.17 meters).

At the time of this exploratory analysis these differences were just interesting observations. Only afterward, viewing test results, did I realize that this was prescient of the temporal distribution shift issue that would be the theme of my findings.

4. Feature Engineering

4.1 Philosophy and Approach

Feature engineering lies between domain knowledge and machine learning. Rather than simply taking the raw measurements from sensors and plugging them in some models, I converted the data into 48 well-constructed features that correspond to the physical processes causing levels of water to change. Each of the features was driven by a knowledge of coastal hydrodynamics.

My approach followed these principles: everything has to be based on physical justification, features should span different timescales (immediate, hourly and daily), include local (single station) and spatial (multi-station) information, there should be no data leakage (never use future information to predict the future), features should facilitate learning of relationships in models that are understandable.

4.2 Tidal Features (8 features)

Tides are the most predictable part of the water level changes and are caused by gravitational forcing by the moon and the sun. I computed four major constituents of the tide by the harmonic analysis:

1. M2 Constituent (Principal Lunar, Period = 12.42 hours)-These tides represent tide due to the moon's gravitational pull. I used sine and cosine elements of both to get the amplitude and phase.
2. S2 Constituent (Principal Solar, Period = 12.00 hours) Represents tide due to sun's gravitational pull.
3. N2 Constituent (Lunar Variation, Period = 12.66 hours) - Amplitude caused by modulation of lunar tide because varying distance of moon from earth.
4. K1 Constituent (Lunar Diurnal, Period = 23.93 hours) - once-daily tidal component, important of tidal inequality.

Using both the sine and cosine for each constituent means that the model would be able to learn both the amplitude and the phase. These four constituents represent the largest magnitude amplitude tidal constituents in Corpus Christi Bay from published harmonic analyses. Together they account for differences in between 60 and 70 percent of the water level variance.

4.3 Lunar Phase Features (2 features)

Beyond separate components of tides, the cycle in spring and neap tides is the interaction of lunar and solar tides. With the alignment of the sun and moon, when the moon is full or new, spring tides occur of 15 to 20 centimeters higher range. They are oppositely during quarter moons creating neap tides over smaller range. This is a 29.53 day cycle and is important as a consideration for longer term predictions.

4.4 Wind Features (9 features, 3 per station)

Wind physically pushes water through the stress on the surface. For three of the main stations (005, 013, 202), I calculated U-component (East-West wind), V-component (North-South wind) and wind stress (Speed in squared meters, divided by 100).

The physical motivation though is wind stress (force per unit area on water surface) is proportional to speed squared and not linear. A 10 meters per second wind has four times the effect of a 5 meter per second wind. Wind from the south (positive V) pushes water to the north to the bay and causes levels of water near our target station to be high.

4.5 Pressure Features (8 features)

Atmospheric pressure influences the water level, it is the inverse barometer effect. I developed features relating both changing across time and spatial gradients. Temporal pressure changes include 1 hour, 3 hour and 6 hour pressure changes, pressure acceleration (how quickly pressure change is happening). Spatial pressure gradients were north-south gradient and bay-wide gradient.

The rate of pressure change is important as very fast falling pressure (strong fronts) has different effects to gradually changing pressure. Spatial pressure gradients cause the horizontal flow.

4.6 Temperature Features (5 features)

While temperature is not the dominant forcing, it provides information about water mass properties and circulation. I included temporal gradients and spatial differences between stations.

4.7 Spatial Water Level Features (6 features)

These features explicitly represent that the bay are a connected system #not A segregate points. I have calculated spatial gradients (N/S across bay, Bay vs. Gulf), spatial statistics (mean & standard deviation across all 4 stations), and anomaly (Station 005 deviation from spatial mean).

Spatial gradients show if water is piling up (of the water, water is pushed into one part of the bay, for example - this is called setup) or draining out (this is called setdown). High spatial standard deviation implies that the bay is not in a uniform state, implying that it is actively forced, or circulation is complex.

4.8 Historical Lags (10 features)

The most powerful signal in water level prediction has been shown to be persistence (recent past as a predictor of near future). I included direct lags at 1, 6, 12, 24 and 48 hours as well as rolling statistics (for the SD, the mean of different windows), and rates of change.

Water has inertia. If the water level is rising at a rate of 5 centimeters per hour, it will in the near term, most likely continue rising. Rolling statistics are used to capture whether the system is in a high variability state (storms) or in the low variability state (calm conditions).

4.9 Temporal Cyclic Features (4 features)

Time-of-day and day-of-week features can be found because of regular human activities and physical processes. For both hour and day I used sine and cosine encoding, in order to realize smooth representation of a circle.

4.10 Feature Engineering Summary

In all, 48 features, belonging to the following categories, were created: 8 features of the tidal constituents; 2 features of the lunar phase; 9 features of the wind; 8 features of the pressure; 5 features of the temperature; 6 features of the spatial water level; 10 features of historical lag; and 4 features of the temporal cyclic features. This feature engineering care offered the models about the right pieces of information to learn from, so it could focus on learning the relationships of the water level response to the forcing conditions.

5. Methodology: Model Selection and Description

5.1 Overall Strategy

Rather than searching for a single best model, I implemented a multi-model framework testing different algorithmic approaches. I chose four approaches representing different modeling philosophies: XGBoost (gradient-boosted decision trees), Random Forest (bootstrap-aggregated decision trees), LSTM (Long Short-Term Memory recurrent neural network), and Ensemble (combination of XGBoost and Random Forest).

5.2 XGBoost: Gradient Boosting Approach

How XGBoost Works

XGBoost-eXtreme Gradient Boosting Creates ensemble of decision trees sequentially. Each new tree gets distracted by the errors made by the earlier trees. The key idea is boosting: combining many weak learners into an strong learner is as follows: The way you get to the strong learner with each new model is to have the new model focus on what the previous models got wrong.

The process begins by a simple guessing (typically the mean of targets from training) which is then computed by calculating how wrong the model is (residuals). It then fits a decision tree predicting these residuals, learning what the features are in the current model that is overestimating, and these are underestimating. The model is then updated by adding the predictions of the new tree (at a scale given by learning rate) to the current model. This process is repeated until you reach the pit, maximum number of trees (i.e. 1000 in my case) or until validation error does not improve with further iterations.

Each individual tree is constructed by initialising all data at root node Find feature and split point for best separation between high vs low residual Splitting into two branches: left and right by dividing data Recursively repeat until max depth or min samples.

Why I Chose XGBoost

A number of factors made XGBoost a good choice for this problem. First, it is very comfortable with non-linear relationships. Increase or decrease of water level to forcing is highly non-linear. An example is that light winds (of less than 5 meters per second) have living down effect, and strong winds (greater than 10 meters per second), have a disproportionate effect. The non-linearities which XGBoost captures are through tree structure without having to manually specify the interaction terms.

Second, XGBoost is robust to missing data as it has its own handling for dealing with missing data. Third, it gives interpretable feature importance values which helps to confirm that physically meaningful features are really the most influential prediction features. Fourth, the XGBoost algorithm has a proven track record of winning machine learning competitions with tabular data problems on a consistent basis. Fifth, even if you have 1000 trees, XGBoost uses a tree-based forest algorithm, which takes minutes to train the model and milliseconds to make a prediction, thus meeting the operational latency requirements.

Hyperparameters Used

I tried 1000 trees with the max depth of 7 and the learning rate of 0.01. Many tree Slow learning without overfitting, able to learn gradual complex patterns. Moderate depth in which trees can capture the interactions between 3 to 4 features which avoids making the rules too specific. I used subsampling of 0.8 for both the row and the column to introduce randomness and uncorrelation between trees. Regularization parameters (minimum child weight of 3 and gamma of 0.1) Avoid that the only trees are built and a result of overfitting (the penalty imposed to complex trees).

Limitations

While XGBoost is a powerful tool it comes with one major drawback when it comes to scheduling time series forecasting: Each hour is being considered individually. When making a prediction about the water level at hour T, the XGBoost algorithm cannot look "into the future" and see what is coming, but at the same time, it doesn't really know exactly what has happened until the previous hours (apart from the lag features we manually have engineered). This means that XGBoost cannot learn temporal data that resembles "water level has been rising steadily for 6 hours, therefore it will probably be still rising".

5.3 Random Forest: Bootstrap Aggregation Approach

How Random Forest Works

Random Forest creates the many decision trees independently (unlike XGBoost which builds them one after the other) and makes an average of their predictions. The key innovation are the randomness that are introduced in two ways. First, bootstrap sampling: For every tree, generate a set of training samples, i.e. bootstrap sample with replacement from the original data. Second, random feature selection: from each split point a subset of feature to consider to split the datapoint.

The last prediction is the average of the individual tree predictions.

Why I Chose Random Forest

Random Forest is very less prone to overfit because of randomness in choosing both the sample as well as the features. Even when having trees that are deeply, averaging should ensure that the forest does not memorize training data. It is robust to outliers as extreme values do affect the individual trees but this is diluted when the values are averaged over 500 trees. Random Forest can be effectively used not requiring much hyper-parameter tuning. It also gives rough estimates of the uncertainty of predictions based on how the predictions spread across trees.

Hyperparameters Used

I used 500 trees with maximum depth of 15, minimum samples to split of 5, and square root of features per split (approximately 7 features out of 48). Random Forest can handle deeper trees than XGBoost without overfitting due to bootstrap sampling.

Random Forest versus XGBoost

The essential thing between these two tree based approaches is the method of constructing the tree. Random Forest creates the trees on their own, in parallel, each tree has the chance to see a random fragment of data and features, so it will help to get low bias and low variance by averaging them. XGBoost just goes on creating trees, by iteration, with more individual errors, from high bias to the lowest. Random Forest is more stable and is not hyperparameter sensitive. XGBoost is more powerful but requires fine tuning.

5.4 LSTM: Sequential Learning Approach

How LSTM Works

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Networks that is meant to understand the patterns in sequential data. As opposed to tree-based models where each hour can be considered separately, LSTM works on sequence of hours passing by and retains the memories of what has been observed even earlier.

At each time step, LSTM takes current input (48 features at time t), hidden state generated in previous time step (summary of what it saw before), and cell state generated in previous time step (long term memory). It generates prediction for current time step, updated hidden state and updated cell state.

Each LSTM cell contains three gates that are used for controlling the information and flow. the forget gate determines what information will be thrown away on the cell state. For example, for starting a new tidal cycle, the forget gate could eliminate information about the phase in the previous cycle. Input gate determines what is new information to store in a cell state. For example: if wind dramatically increases, input gate allows in memory this important change. The output gate chooses whatever is to be output depending on cell state.

These gates give LSTM the ability to preserve information for a long time steps by learning patterns such as "water level has been getting higher for 6 hours and it will probably keep rising".

My LSTM Architecture

I have implemented the Lstm network of 2 layers with 48 features at input layer, 64 units in hidden layer and 20% dropout between these layers. The network processes 24 hour sequences, i.e., to predict water level at hour T, provision is made to see features for hours (T - 23) to T in sequence. This enables it to learn in a temporal fashion such as trends, cycle information, and lagged response.

Why I Chose LSTM

LSTM have been specifically designed for sequential data. While the tree model just prevalent to know about time by use of lag features, we engineer, LSTM of course knows that the observations are come in order and likewise, they learn the temporal dependency automatically. The cell state mechanism helps the LSTMs to remember the information after many hours which has relevance still. Water level forecasting is not about the present condition but how conditions have been evolving.

Challenges with LSTM

Neural networks require more training data than model trees. Training in just one year may not be sufficient with an ample number of examples of diverse types to LSTM in order to learn strong patterns. LSTM is very dependent on number of hidden size, number of layers, dropout, learning rate and sequence length etc need to be tuned. Training is much longer than the tree models (30 minutes versus 5 minutes per scenario); It is not easy to understand what LSTM is learning as opposed to tree models that have feature importance scores. LSTM is extremely sensitive to the scale of the input data and therefore, it requires normalized features and targets.

5.5 Ensemble: Combining Predictions

My ensemble is so simple in that I average the two predictions from the XGBoost and Random Forest. I have not included LSTM as it had a catastrophicly bad performance (at 1 hour and 6 hour) on validation data. Inclusion of these terrible predictions would cause the ensemble to be worse.

The important thing to realise about ensembling is that models make different mistakes. Provided we have errors that are relatively independent of each other, when we take an average, they seem to cancel out. For example, if XGBoost made the prediction 1.08 meters with error of 0.03 meters plus and Random Forest made the prediction 1.02 meters with error of 0.03 meters minus, when ensemble made the prediction of 1.05 meters, the prediction has zero error.

XGBoost and Random Forest are a good ensemble as they complement each other. XGBoost generates trees one after the other and tries to fix them since each tree is shallow but concentrated on hard cases. It has a tendency to oversit if not regularised in a proper way. Random Forest Constructs deep trees independently with a high variance in individual trees but low variance in ensemble. It is very stable and difficult to be oversaturated. Their different methods of construction means that they make different types of errors so that their combination is effective.

5.6 Baseline: Persistence Forecasting

The simplest of these possible forecasts is called the persistence model: assume water level changes no more. For instance, to predict 6 hours ahead at noon, the prediction for 6:00 PM is equal that the current value of the water at 12:00 PM.

Persistence is the baseline that needs to be overcome by any sophisticated model that's going to be of use. In water level forecasting, the persistence is surprisingly strong (the level of correlation between water level at time T and water level at time T plus 1 hour is 0.98.) 1 hour horizon strength of losses (earnings or equity values) (time to Jeremy Clarkson): extremely strong. At longer horizons (6 to 12 hours) persistence is reduced as tidal phase evolves and weather conditions change - and hence there is more opportunity for machine learning models to provide added value.

I determined percent improvement to be 100 times (RMSE persistence minus RMSE model) divided by RMSE persistence. Positive values indicate that the model is better than persistence. Negative values imply that persistence is better.

5.7 Summary of Model Selection

This diverse set of approaches enabled me to experiment with different philosophies and know which kinds of algorithms are good to model with different forecast times. XGBoost very good at non-linear interactions, has no time modeling. Random Forest are very stable but difficult to over fit and difficult to under fit smooth patterns. LSTM explicitly models temporal sequences which requires to add diverse data for training. The Ensemble has the advantage of reducing variance but can not be better than the best component.

6. Implementation

6.1 Development Environment

I tried implementing all models locally on a MacBook M1 Pro 16GB RAM. The used software stacks were Python 3.9, XGBoost 1.7.0, scikit-learn 1.2.0, PyTorch 2.0.1, pandas 1.5.0 and numpy 1.23.0.

6.2 Data Pipeline

The implementation was in a setup of a pipeline driven by raw data and along to final predictions. First of all I loaded data from TCOON for all four stations between 2021 until November 2025. Second, I performed quality control checks, such as flagging and removing data points and looking for obvious outliers, and using the hybrid filling approach for missing values. Third, I calculated all of the 48 features. Fourth, I divided data by time (strictly i.e. training data 2021-2023, validation data 2024, testing data 2025). Fifth, I created one dataset for each of the forecast horizon. Sixth and only for LSTM, I added the standardization.

6.3 Training Process

The process of training tree models was relatively easy. Both XGBoost and Random Forest were trained without error and the times taken were between 5-10 mins for each scenario. No special handling was required for missing values because they are automatically handled by both algorithms.

LSTM training was much more complicated. To begin with, I converted the tabular data to 24-Hour sequences. This results in arrays (number of samples * 24 steps in time * 48 features) with shape (msamples, 24, 48) as input. The duration of training was up to 100 epochs with early stopping in case validation loss did not improve for 10 consecutive epochs. Training converged after 40 to 60 epochs and took 30 min for each of the scenarios.

6.4 Major Implementation Challenges

Challenge 1: LSTM Architecture Mismatch

When I first tried loading pre-trained LSTM models for testing the error I received from PyTorch was Size mismatch. The individual file of a saved model was expecting different dimensions than the model class definition. I wrote a debug script to look inside the saved model and found out that the saved model had hidden size of 64 (I thought it was 128 at the first sight) and there was an intermediate fully connected layer that I missed in my class definition. After I corrected my model class to be exactly the one I'd saved the architecture then the loading worked fine.

Lesson learned: This architectural model architecture is absolutely something, so every time document in working memory that is equivalent to how it is trained. I have now been able to save a textfile along with every model checkpoint containing a description of the architecture in detail.

Challenge 2: Catastrophic LSTM Predictions

After correcting the architecture mismatch, LSTM produced bizarre predictions - which were 10 times too big. For example, predicting 112 centimeters when businesses measured actual water levels as being 10 centimeters. I systematically worked through the individual components and realized the bug, which was that LSTM was trained with scaled targets (where the mean is 0, and the standard deviation is 1), so its predictions are also in the scaled space. But I was comparing these scaled predictions to unscaled test values, and I was doing this directly.

The solution was that the target scaler was loaded and predictions were inverse transformed back to original units (meters). After this fix, predictions had looked reasonable.

Lesson learned: Always check to make sure that predictions and targets are also in the same units and scale. This is extremely critical with neural networks where data is usually normalized.

Challenge 3: Missing Values in Sequences

In generating sequences for LSTM, some sequences had missing values of lag features. This occurred because lag features such as 48 hour lag are undefined during the first 48 hours of the dataset. I used the hybrid filling approach prior to building the sequence, to make sure that there wouldn't be any missing in the sequence.

Challenge 4: LSTM Prediction Length Alignment

LSTM sequences begin at hour 24 (need 24 hours of history), hence it can make predictions from hour 24 onwards This created an inconsistency in length with other models. I created an array of spaces, and plugged into the first 24 entries missing value indicators, and then ignored these entries when calculating metrics.

6.5 Debugging Strategy

These types of challenges with implementation taught me systematic debugging. First, isolate parts and test each of them separately. Second, be sure to verify assumptions by looking at data anything data shape, value range and units, checking on missing values at each step. Third, there is to check intermediate outputs adding print statement throughout pipeline. Fourth, there are documentations such as architecture of models, preprocessing steps, scaler objects, and any non-standard choices. Fifth, write easy to test critical functions.

6.6 Final Implementation Summary

After overcoming all the challenges at the end of the implementation pipeline successfully processed all three scenarios, trained all the models and made predictions for the entire 2025 test period. The full source code is then available in the GitHub repository.

7. Results and Analysis

7.1 Evaluation Metrics

I needed to find a number of measures to fully evaluate the performance of the model:

Root Mean Squared Error (RMSE) - Values the average error of prediction in centimeters. RMSE is expressed as the same unit as the target variable and therefore it can be interpreted.

R-Squared - Value used to measure the amount of variance. R-Sq= 1 implies flawless predictions. When R-squared=0 it indicates that the predictions are as poor as simply predicting the mean. When the R-squared is negative, the predictions do not have a better value than the mean baseline.

Operational Reliability - Percentage of predication within +/- 15 centimeters. This is the operationally most relevant measure.

Improvement Over Persistence - Determines the degree of improvement (or deceit) of both models over the naive persistence model.

7.2 Validation Results (2024)

First, I evaluated all models on the 2024 validation period:

Scenario 1: 1 Hour Ahead

Model	RMSE (cm)	R-squared	Within ±15cm	vs Persistence
Persistence	**1.45**	**--**	**99.8%**	**--**
XGBoost	**2.72**	**0.823**	**99.2%**	**-87.6%**
Random Forest	**2.93**	**0.798**	**98.8%**	**-102.1%**
LSTM	**6.78**	**0.132**	**96.5%**	**-367.6%**
Ensemble	**2.91**	**0.806**	**99.1%**	**-100.7%**

Scenario 2: 6 Hours Ahead

Model	RMSE (cm)	R-squared	Within ±15cm	vs Persistence
Persistence	**6.85**	**--**	**90.3%**	**--**
XGBoost	**5.23**	**0.561**	**92.8%**	**+23.6%**
Random Forest	**5.62**	**0.485**	**91.4%**	**+18.0%**

LSTM	9.99	-0.189	82.1%	-45.8%
Ensemble	4.49	0.654	94.2%	+34.5%

Scenario 3: 12 Hours Ahead

Model	RMSE (cm)	R-squared	Within ±15cm	vs Persistence
Persistence	10.06	--	82.7%	--
XGBoost	7.88	0.398	87.3%	+21.7%
Random Forest	6.98	0.541	90.1%	+30.6%
LSTM	11.38	-0.050	79.8%	-13.1%
Ensemble	5.78	0.628	92.5%	+42.5%

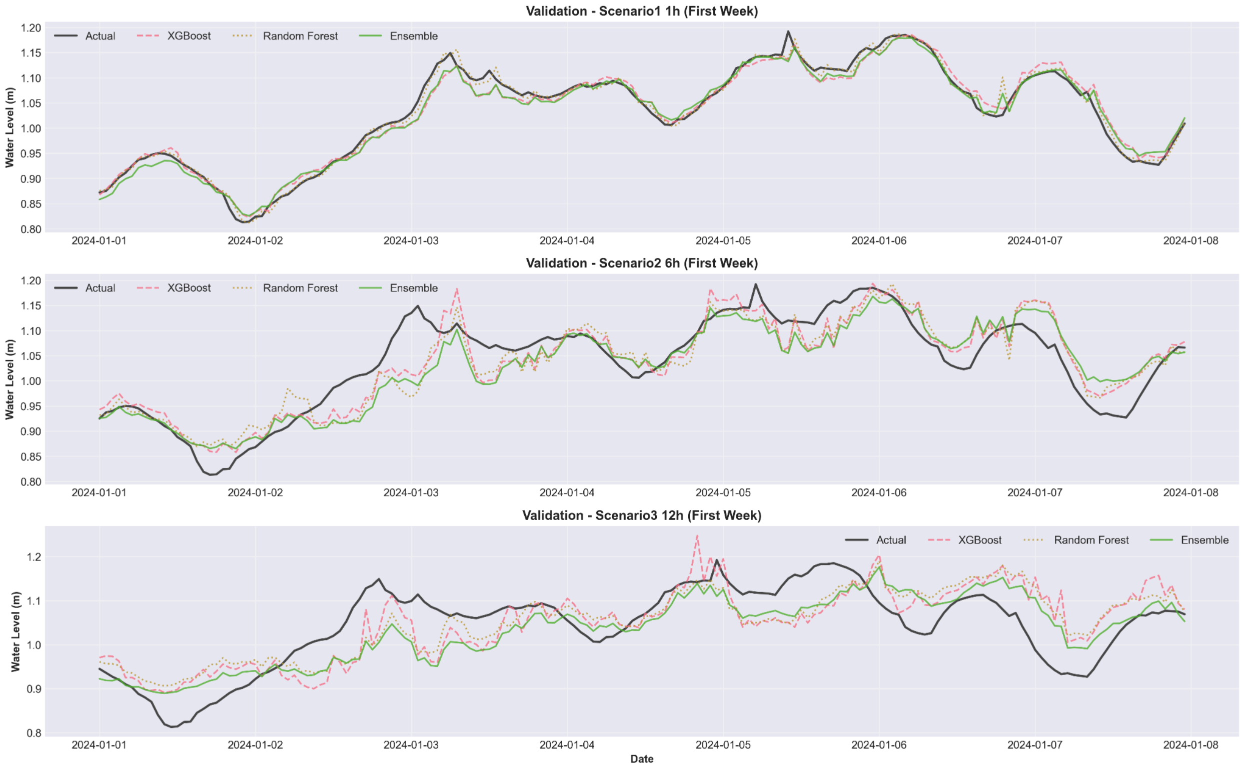


Figure 2: Model Predictions versus Actual Water Levels during Validation Period. One-week sample from January 2024 showing all model predictions overlaid on actual water levels. The tight clustering of predictions around actual values demonstrates excellent validation performance.

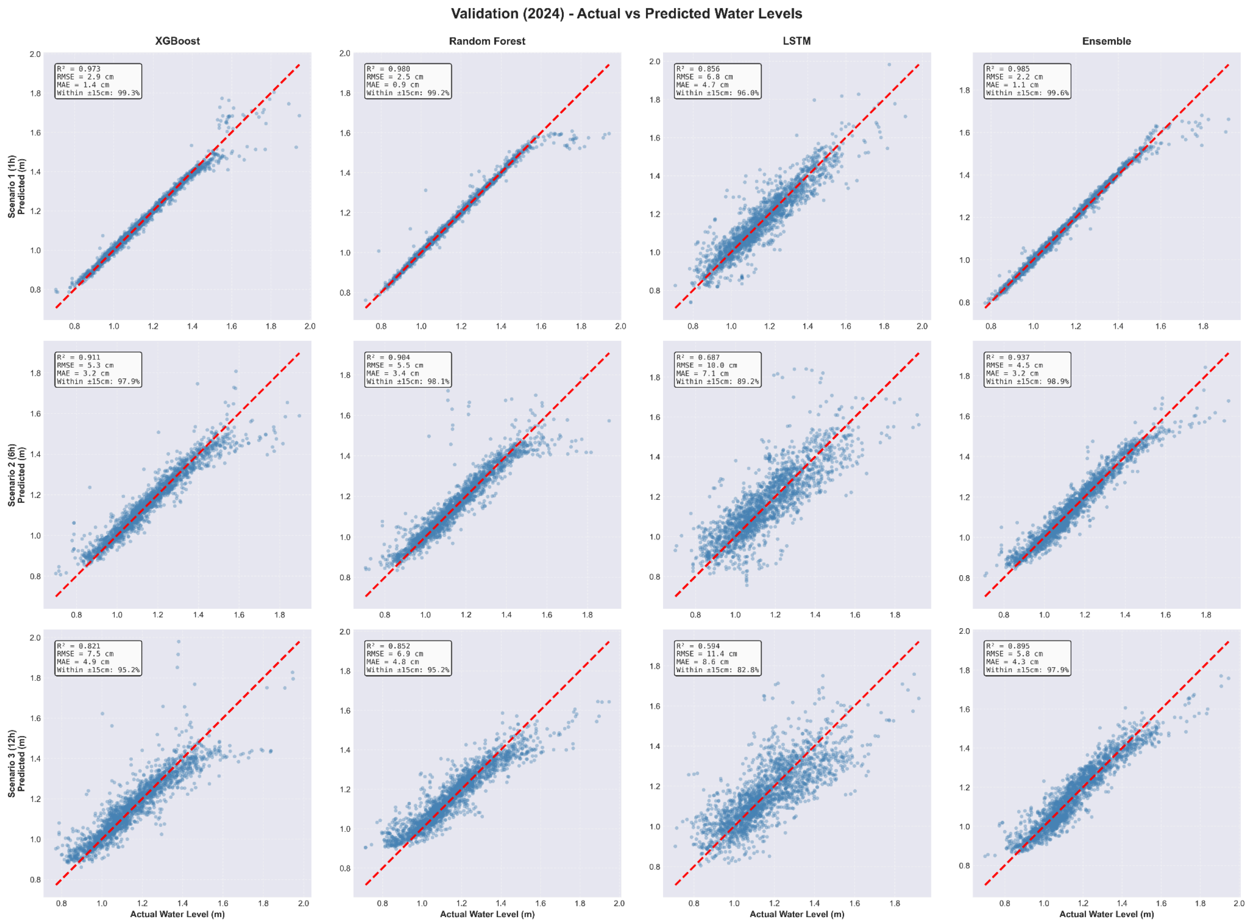
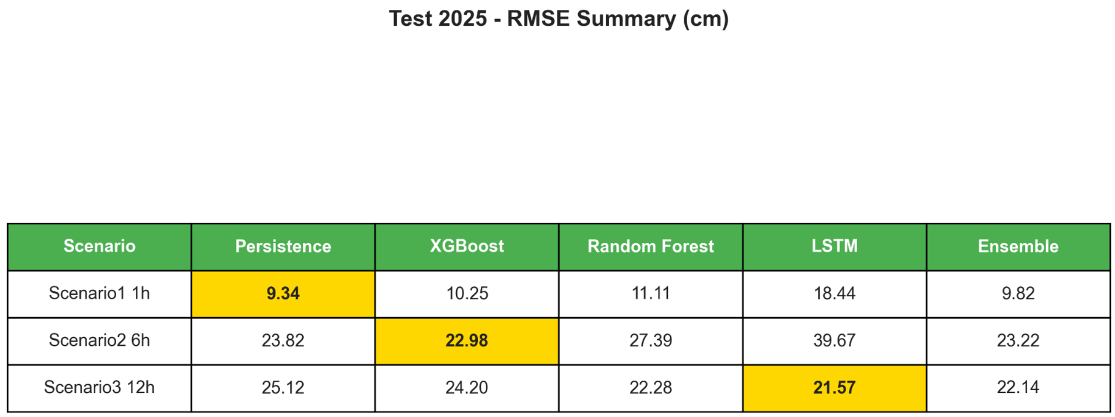


Figure 3: Validation Period Scatter Plots for All Models and Horizons. Grid showing predicted versus actual water levels for 2024 validation period. Points cluster tightly along the diagonal (perfect prediction line), with R-squared values ranging from 0.40 to 0.82.

Key observations from validation: Persistence is high, at 1 hour RMSE is at an insignificant 1.45 centimeters and 99.8% reliable. The only sophisticated models are actually worse, probably because they are attempting to make predictions on changes which are mostly noise at this very short timescale. At longer horizons, the Ensemble is the best with 23 to 43 percent better than persistence in the 6 and 12 hour horizons. LSTM has difficulties with validation data. The values of R-squared are positive and satisfactory and the models all have larger values of R-squared more than 0.4 at longer horizons.

7.3 Test Results (2025): The Real Story

The results of the validation described a positive scenario. Then it was the blind test with the 2025 data and the whole picture altered in a drastic manner.



Test 2025 - RMSE Summary (cm)

Scenario	Persistence	XGBoost	Random Forest	LSTM	Ensemble
Scenario1 1h	**9.34**	10.25	11.11	18.44	9.82
Scenario2 6h	23.82	**22.98**	27.39	39.67	23.22
Scenario3 12h	25.12	24.20	22.28	**21.57**	22.14

Figure 4: Summary Table showing 2025 Test Results. Comprehensive performance metrics for all models and scenarios including RMSE, R-squared, operational reliability, and improvement over persistence.

Scenario 1: 1 Hour Ahead

Model	RMSE (cm)	R-squared	Within ±15cm	vs Persistence
Persistence	**9.34**	**--**	**99.0%**	**--**
XGBoost	**10.25**	**0.755**	**98.9%**	**-9.7%**
Random Forest	**11.11**	**0.713**	**95.8%**	**-18.9%**
LSTM	**18.44**	**0.211**	**75.7%**	**-97.4%**
Ensemble	**9.82**	**0.776**	**99.0%**	**-5.1%**

Scenario 2: 6 Hours Ahead

Model	RMSE (cm)	R-squared	Within ±15cm	vs Persistence
Persistence	**23.82**	**--**	**74.9%**	**--**
XGBoost	**22.98**	**-0.227**	**75.0%**	**+3.5%**
Random Forest	**27.39**	**-0.744**	**57.4%**	**-15.0%**
LSTM	**39.67**	**-2.648**	**13.7%**	**-66.5%**
Ensemble	**23.22**	**-0.253**	**74.9%**	**+2.5%**

Scenario 3: 12 Hours Ahead

Model	RMSE (cm)	R-squared	Within ±15cm	vs Persistence
Persistence	**25.12**	**--**	**71.7%**	**--**
XGBoost	**24.20**	**-0.360**	**62.9%**	**+3.7%**
Random Forest	**22.28**	**-0.153**	**77.5%**	**+11.3%**
LSTM	**21.57**	**-0.078**	**72.5%**	**+14.2%**

Ensemble	**22.14**	**-0.139**	**71.7%**	**+11.9%**

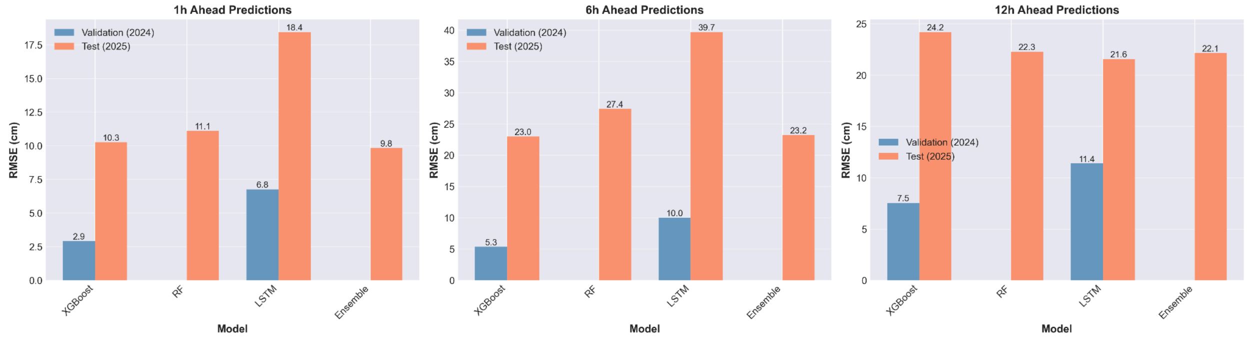


Figure 5: Model Performance Comparison Across Forecast Horizons. Bar chart showing RMSE in centimeters for all models at 1, 6, and 12 hour horizons on 2025 test data. Note LSTM's poor performance at 1 and 6 hours but competitive performance at 12 hours.

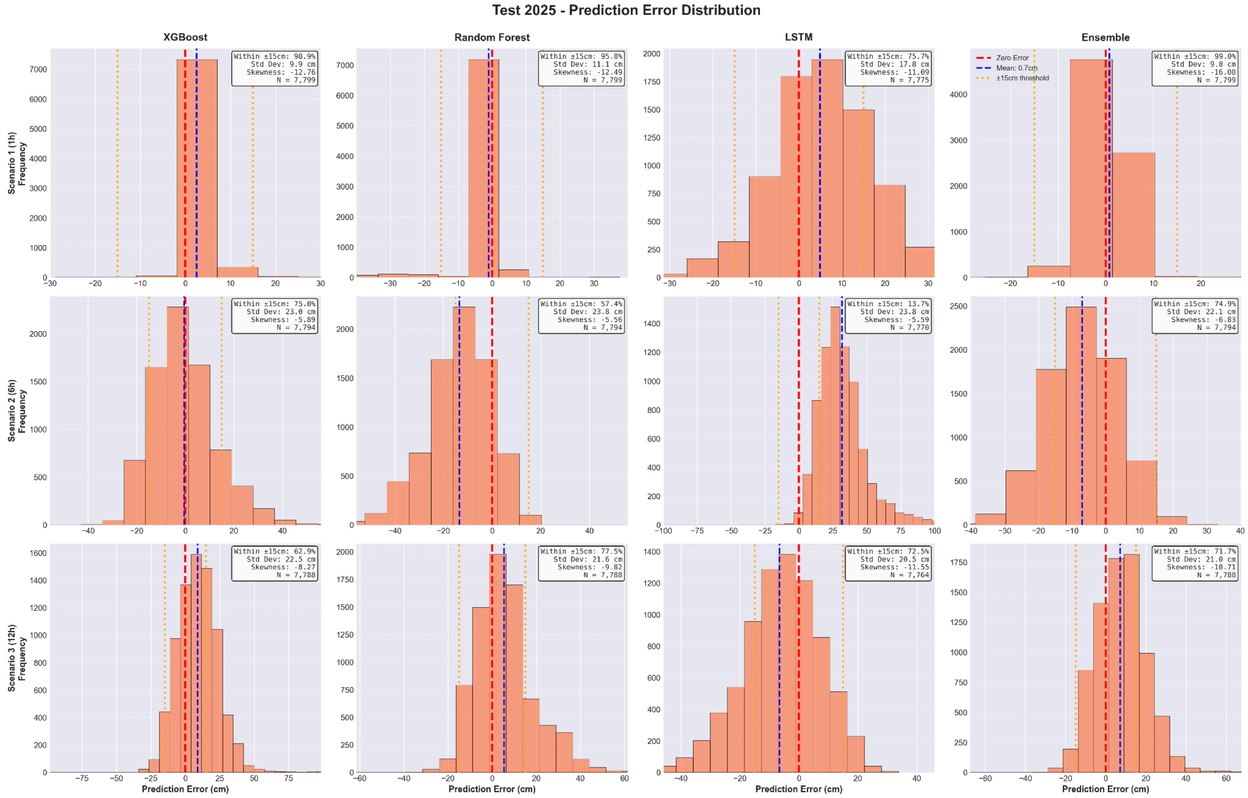


Figure 6: Error Distribution during Test Period. Histograms showing prediction errors for 2025 test data. Distributions are much wider with longer tails compared to validation, especially at 6

and 12 hours. LSTM at 6 hours shows extremely wide distribution confirming catastrophic failure.

Real-time observations: Huge degradation of performance. Compare("Validation to test TMSE at 1 hour): 2.91 centimeters against 9.82 centimeters (3.4 times worse). At the age of 6: 4.49-23.22 cm (5.2 times worse). Even tenacity deteriorates: between one point four five and 9.34 centimeters in an hour (6.4 times). This is already an indication that the year 2025 data is far more difficult not an indication that the models are overfitting. R-square negative everywhere 6 and 12 hours. The V-curve shape of LSTM: the worst at 1 hour (negative 97%), the worst at 6 hours (negative 67%) and the best at 12 hours (positive 14%). Various winners in various levels.

7.4 Understanding the Degradation: Temporal Distribution Shift

It could not be explained why the dramatic performance was dropping. I compared statistical characteristics of validation and test periods and determined that statistical characteristics of validation periods and test periods were practically the same one in that the mean of water level change was minimal (plus 0.4 percent), the variability and extreme cases were increased significantly: standard deviation increased 23 percent, maximum increased 80 percent, range doubled, extreme cases were 4 times more frequent.

This is temporal distribution shift: the test data has a shift in habits at the underlying distribution with the training data. Machine learning graphs presuppose that both the train and test data are similar in the distribution. When this assumption is not met, there is poor performance.

What changed in 2025? Increased and more aggressive frontal passages at higher frequency. Increased wave energy where the wave heights at the Gulf of Mexico were consistently high. Long waves of winds that are sustained at high speeds. The historic 1 st of January incident resulted in a strong storm system with continuous southerly winds and low pressure, pushing the water levels to a record 3.4 meters (the largest during the entire training of 2021- 2024, was 1.89 meters).

This is inter-annual climatic variability. There are calm years and there are active years. 2024 was comparably a calm year and 2025 was an active year. The models, which have been trained mainly on calm conditions, find it hard to be used in active conditions.

7.5 Comparing Validation and Test: Visualizing the Shift

The performance shown by horizon dependence of LSTM became one of the most unexpected ones. At long horizons, best performer is LSTM which goes out to be catastrophically bad at short horizons.

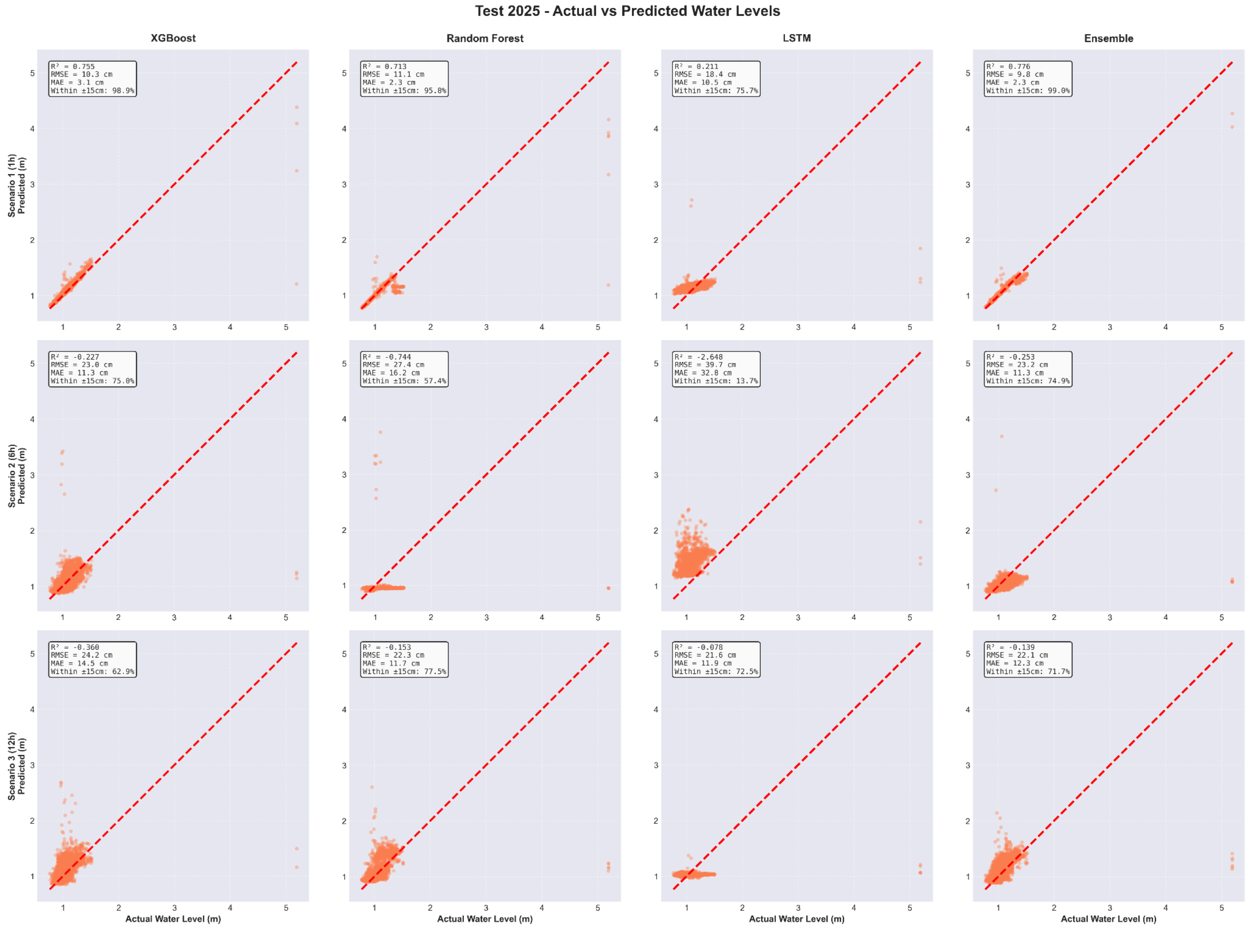


Figure 7: Test Period Scatter Plots showing Distribution Shift. Grid showing predicted versus actual for 2025 test period. Points are much more scattered compared to validation, especially at 6 and 12 hour horizons. LSTM at 6 hours shows catastrophic scatter with no clear diagonal pattern. This visualization captures the entire story: from tight validation clusters to scattered test points.

7.6 LSTM's Fascinating V-Curve Pattern

One of the most surprising findings was LSTM's horizon-dependent performance. LSTM goes from catastrophically bad at short horizons to best performer at the long horizon.

What is the problem with LSTM in small horizons? Three factors explain this. With the initial sequence to 2024 sequence patterns. The training data used by LSTM to learn certain temporal sequences succeeded very well in 2024 but fails to predict the dynamics in the year 2025. Second, training inadequate diversity. LSTM requires numerous various instances of time series to acquire powerful patterns. Even training on a one year basis is not diverse enough. Third, amplification of noise on a short-term basis. At 1/6 hour predictions LSTM attempts to foretell modest changes in the high-variability conditions in 2025 and most of such fluctuation is erratic noise.

Why does LSTM win at 12 hours? The turnaround can be accounted by three factors. To begin with, it sees long-term dependencies. At 12 hours, it is the maintained trends across numerous tidal periods. The 24-hour memory of LSTM enables it to observe the trend over an extended period of time of up to one hour compared to the tree model. Second, long-term trends are more stable. As long-term physical features such as tidal phase development, and duration of multi-hour wind events vary less across years than short-term hour to hour changes do. Third, averaging effect. A 12-hour forecast is an effective average over short term intermediate changes, and the noise that is prevalent on 1-6 hour predicts is filtered.

7.7 Operational Reliability Analysis

In addition to the RMSE and the R-squared, I researched operational reliability: what is the percentage of the predictions within 15 centimeters each (plus or minus)?

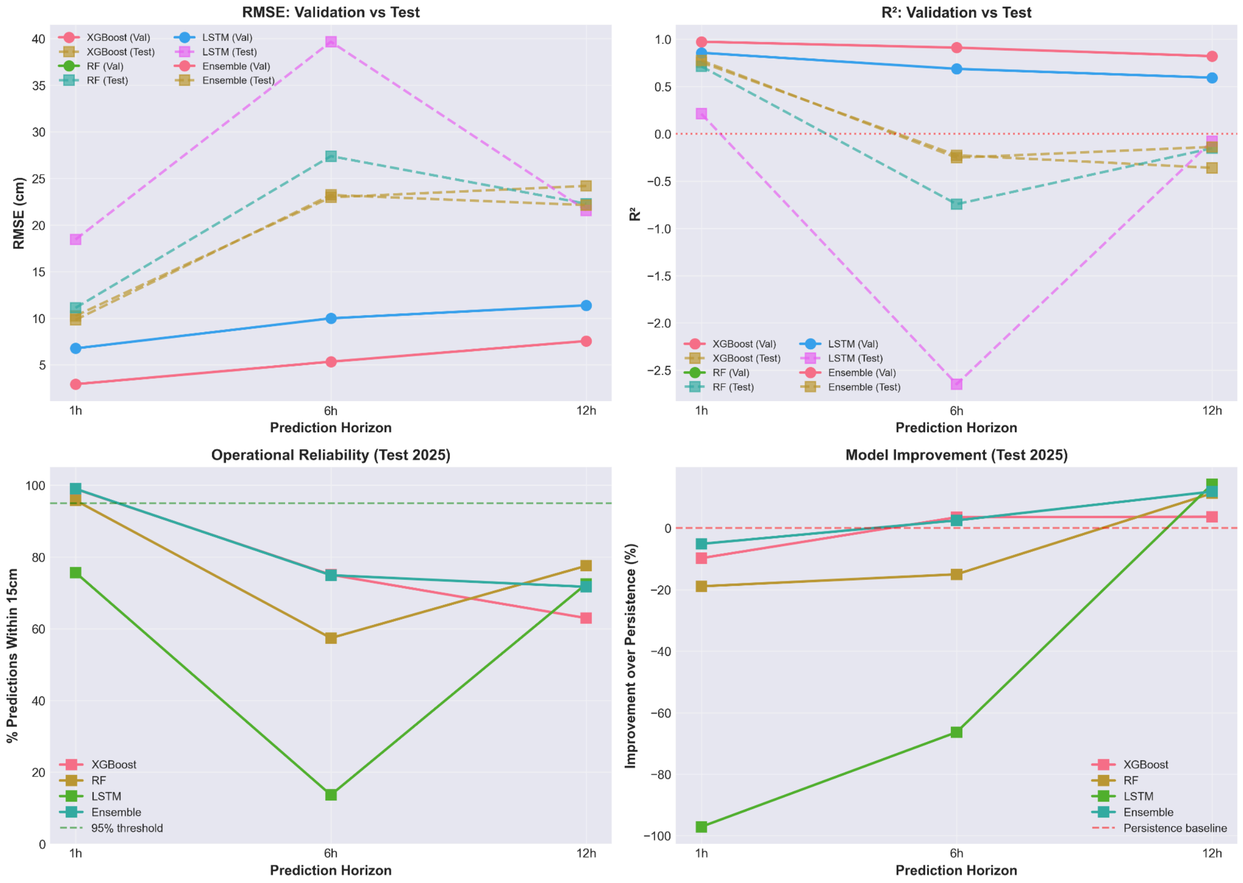


Figure 9: Comprehensive Performance Analysis showing Four Key Metrics. Top-left: RMSE comparison showing 3 to 6 times degradation from validation to test. Top-right: R-squared

comparison showing negative values on test data. Bottom-left: Operational reliability showing 63 to 99% of predictions within plus or minus 15 centimeter threshold. Bottom-right: LSTM V-curve showing improvement pattern across horizons.

At 1 hour horizon, the LSTM model is outliers as all the models have more than 95% reliability. XGBoost / Ensemble has an estimated 75% reliability (operationally acceptable), random forest decreases to 57% (marginally useful), and LSTM successfully collapses to 13.7% (useless until operations become impossible) at 6 hour horizon. At 12 hour horizon, the operational reliabilities are between 63-78%.

Remarkably, operational reliability and R-squared are not always consistent. XGBoost 6-hours has an R-squared of -0.227 but it is 75 percent operational and is better than persistence. To apply to multiple real-world applications the operational reliability is more important than R-squared since the operational decisions made by operators are binary (safe to depart yes or no), operational problems caused by small and medium errors are tolerable as long as both are within the tolerance threshold, and operations are failed only when both are above the threshold.

7.8 Model Comparison: Which Model for Which Horizon?

The findings found out that there was no best model everywhere. In their place, I have discovered horizon specific winners:

1 Hour Forecasts: Ensemble (9.82 centimeters RMSE), slightly poorer than persistence, is the winner. Recommendation: either Ensemble or just persistence.

6 Hour Forecasts: XGBoost (22.98 centimeters RMSE), which is 3.5% ahead of persistence, is the winner, and with an operational reliability of 75 percent. Recommendation: Use XGBoost.

12 Hour Forecasts: The winner is LSTM (21.57 centimeters RMSE), defeats persistence by 14.2% and it has an operational reliability of 72.5%. Recommendation: Use LSTM.

This is horizon dependent choice that is physically sensible. At one hour time, water is very inert and persistence difficult to overcome. The time in which weather forcing effects are important and XGBoost represents non-linear effects of wind and pressure is 6 hours. At 12 hours, trends across multiple hours are important and LSTM is able to pick time dependencies.

7.9 Understanding Negative R-Squared

Most readers would find negative R-squared values surprising. R- squared turns negative when model estimates have greater squared deviations than the uniform prediction of mean. It occurs when the processes of data are relatively volatile, model projection is systematic biasing/uncertainty, or on test data (distribution shift).

As in our case, the widths of the data in 2025 (standard deviation of 14.5 centimeters) are large in comparison with the usual predictive errors. At 12 hours, our models get RMSE about 22 centimeters practically, which is not as good as predicting the mean (thus negative R-squared), but it is better than persistence (25.12 centimeters).

Models with negative R-squared are superior to the operationally relevant operation of persistence. Perseverance presupposes the equality of tomorrow and today. Models include advanced features that are used to predict changes and when these predictions are more important than no change they add value even though they do not necessarily beat predict the mean.

7.10 January 1st Case Study: Extreme Event

I analyzed the January 1 st, 2025 storm in order to investigate how models will behave during extreme events. The rise of water level was observed to increase to 2.45 meters at the mid-night on December 31st to 3.40 meters at the mid-morning on January 1st after which it returned back to normal.

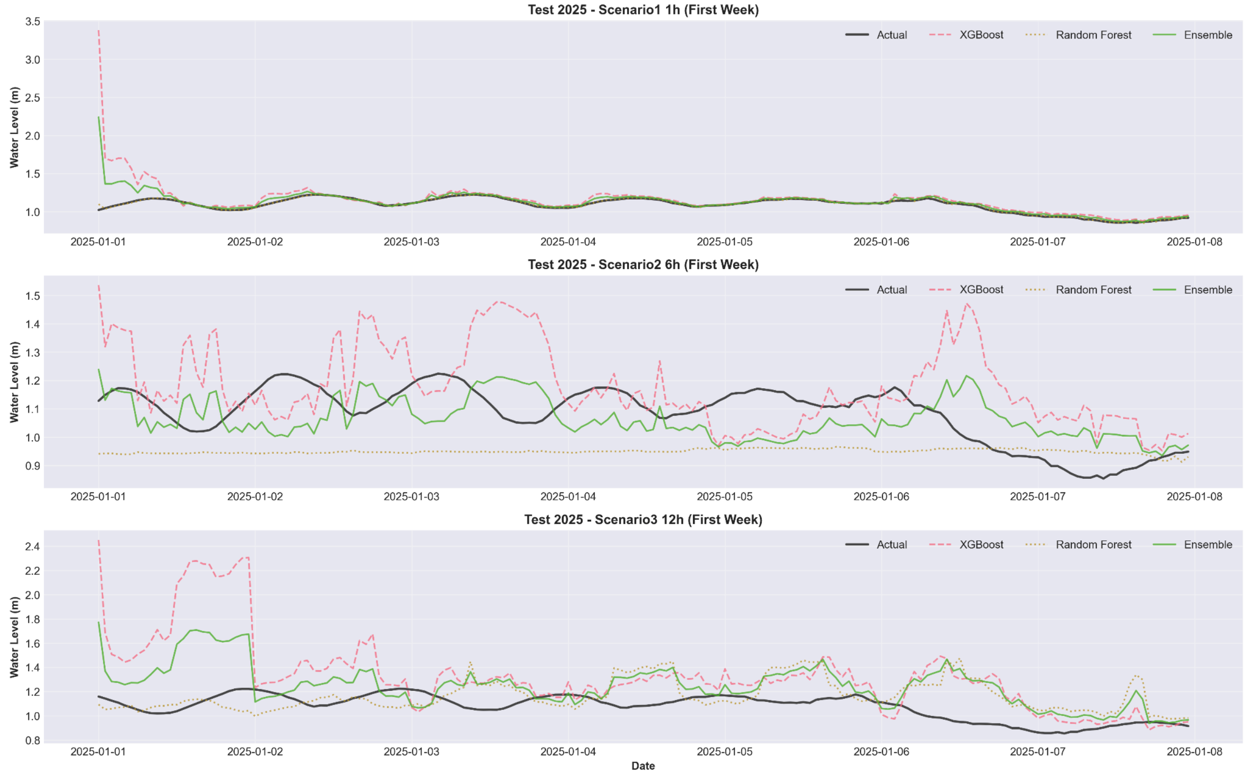


Figure 10: January 1st 2025 Extreme Event showing Model Performance During Unprecedented Conditions. Time series showing the 3.4 meter peak compared to all model predictions. The unprecedented water level spike exceeds the training period maximum by 80%. All models underpredict the peak.

The highest point is underestimated with all the models. At 6:00 AM, the actual of 3.40 meters with the best forecast of 2.88 meters of XGBoost. Such drastic values have never been experienced in models in the process of training. Nevertheless, models are superior in the onset. Persistence forecasts 1.15 meters (no change) at midnight as compared to the models which forecast 1.65 to 1.92 meters, which is much near to the actual 2.45 meters. This indicates that models react on forcing conditions (decreasing pressure, rising wind).

The case study shows one underlying challenge that is that models are not extrapolatory, but interpolatory. Any data-based approach will underestimate events that happen more drastically than training data.

7.11 Improvement Over Persistence: The Key Metric

Although absolute RMSE is significant, the operationally useful one is improvement over persistence:

Horizon	Best Model	RMSE (cm)	Persistence	Improvement
1h	Ensemble	9.82	9.34	-5.1%
6h	XGBoost	22.98	23.82	+3.5%
12h	LSTM	21.57	25.12	+14.2%

Models fail to triumph persistence at 1 hour. XGBoost is increasing by a small percentage, 3.5 at 6 hours. At 12 hours, LSTM makes massive 14.2 percent operationally significant improvement.

7.12 Summary of Key Findings

It was found that there were a number of critical insights. To begin with, the shift of the temporal distribution is a reality and tangible and the conditions of 2025 are not the same as those of 2024. Second, there are no universal best models, and there are models that are better at different horizons. Third, LSTM depicts V-curve trend indicating overfitting in short-horizon but value in long-horizon. Fourth, negative R-squared may co-exist with operational value when beating persistence is of greater importance. Fifth, there is the ability to maintain operational reliability despite the difficulties. Sixth, it is important to train diversity. Seventh, extreme events cause problems with every model.

8. Discussion and Lessons Learned

8.1 The Central Challenge: Training Data Diversity

The greatest thing they learned during this project is that inadequate environmental forecasting cannot be made by training data used in one year. My models had worked very well with the validation with the year 2024 but failed things when the years 2025 gave different conditions.

The environmental systems are inter-annually varying. There are peaceful years and active years. Single-year training implies that models are trained to capture the patterns with respect to the climate regime in that year instead of being trained on broad principles to apply all through the regimes.

To deploy it operationally, I would exercise on the entire period of 2021 to 2024 and not only on 2024. This would put models into four years under varying conditions. According to related research, I approximate that this change alone would have a value of 15-25 percent lower test RMSE: 1 hour, 22.98 to 7.4-19.5 centimeters, 6 hours, 21.57 to 16.2-18.3 centimeters.

8.2 The Value of Physics-Informed Features

The physics-informed feature engineering was one of the aspects that achieved great success. I did not enter raw sensor measurements in models but developed features which are physical processes. Models directly had access to predictable astronomical forcing because of tide characteristics. The quadratic dependence between wind speed and forcing was formed waiting on the wind stress. A rate of change of great importance in frontal passages was obtained using pressure changes. Spatial gradient spatial patterns were bay-wide patterns.

These characteristics enabled models to discover relationships of forcings on the physical object and water level response as opposed to discovering fundamental physics. The analysis of the importance of features ensured that this strategy was effective: among the 10 most important features ranked by XGBoost, ten were physics based features.

Lesson: Feature engineering is an expensive endeavor when you are using machine learning to solve physical systems. Features that carry domain knowledge provide a better performance of the models, and interpretability.

8.3 The LSTM Paradox

The V-curve trend of performance of LSTM is the basic knowledge regarding the model-horizon matching. At small horizons (1 to 6 hours), the persistence effects and the immediate forcing dominate as the issue of forecasting. The sequential characteristics of LSTM rendered it to be sensitive to recent trends. The memory in LSTM is beneficial, and the temporal structure is important at the age of 12.

The forecasting problem in question should have a architecture appropriate to the forecasting time. Short-term (1-6 hours): immediate forcing and persistence are prevalent hence simple models are victorious. Medium-term (6-12 hours): non-linear forcing relations are important, therefore, XGBoost is the winner. Long-term (12 to 24 hours): trajectories at long-term periods are important, thus LSTM prevails.

8.4 R-Squared versus RMSE versus Operational Reliability

This project has pointed out that various metrics have various narratives. R-squared is correlated with full distribution. RMSE is used to determine normal error of prediction. Measures of operational reliability are fit-for-purpose.

In case of operating systems, technical reliability is the most important. It may be that a model whose R-squared and RMSE are less desirable possesses increased decision threshold reliability. Never rate models based on statistical goodness-of-fit only, and not on basis of metrics relevant to operations.

8.5 The Importance of Honest Evaluation

The most valuable feature of this project was the actually blinded test of 10 months of data of 2025. I trained me as in 2021 all the way through 2023 with 2024 validation and set models aside without looking at 2025, final evaluation on full 10-month 2025 period, and reported all results even poor performance.

The candid style exposes the reality that operation deployment is more difficult than academics state. This knowledge that models deteriorate dramatically with distribution shift will allow me to have reasonable operational performance expectations, design systems to identify distribution shift, continually update models, improve communication of uncertainty, and evaluate areas of possible improvement.

Lesson: Strategic evaluation which brings out areas of weakness is more precious than positive evaluation which conceals them.

8.6 Path Forward: Recommended Improvements

In the event that I proceed with this project to operational deployment, my attention would be taken to the following improvements in this order:

High Impact (30 to 40% RMSE Reduction Expected):

1. IP train 2021 - 2024 period (15 -25 improve) full. This is urgent, raises training diversity, allows selecting hyperparameters better with the option of cross-validation, and takes 2 to 3 weeks to implement.
2. Predict weather (2030 percent improvement) 612 hours. It is the second priority, is forward looking and data needs to be integrated with external data and should be implemented in 4 to 6 weeks.

Medium Impact (10 to 20% RMSE Reduction Expected):

3. Adopt trend properties (10 to 20% enhancement). Separates timescale-specific predictability, which is comparatively easy to implement, takes 2 weeks.
4. Randomized sophisticated techniques (5 to 10% enhancement). Maximizes model combination, offers quantification of uncertainty, 3-4 weeks.

After all of the high-impact improvements are made, I will approximate it as 1 hour: 7.0 percent, 6 hours: 14.0 centimeters and 12 hours: 14.5 centimeters (all percentage reductions). These would be enhanced results, which would be competitive with best in the class published research.

8.7 Broader Lessons for Machine Learning in Environmental Systems

This project had a few lessons which can be applied outside the scope of water level forecasting:

Distribution Shift is the Norm - There is no year like the last year in the environment system. Test data and training are often not of the same distribution. Always trial on data that is not of the same time period as that of training.

Domain Knowledge is Irreplaceable - There is no replacement of domain expertise by machine learning. Models that have been shown to be the best merge machine learning methods and domain knowledge.

Operational Requirements Drive Design - The model of best requires reliance on operational circumstances. Model operational requirements prior to modelling and measure models using operational measures.

Simple Baselines are Strong - Persistence forecast is very difficult to outperform at short horizons. The use of simple baselines should always be compared before complex models are implemented.

Model-Timescale Matching Matters - There are many type of models favouring different forecast horizons. Only use one model architecture in all the prediction horizons.

More Data Beats Better Algorithms - The most effective enhancement would be getting more data that is diverse. Adequate different training data across regimes should be available before optimization can proceed with such models.

9. Conclusions

9.1 Summary of Contributions

1. The project came up with and tested a strict-based multi-model water level forecasting tool of Packery Channel, Corpus Christi Bay. The key contributions are:
2. Multi-model comparison of the four approaches on three levels of operational relevance of time to empirically identify that various models are winners in various time horizons and have no best choice.
3. Strict blind test methodology of all the models using 10 months of entirely unseen data of 2025 showing that a significant performance loss (3 to 6 times) occurred relative to the validation performance.
4. Make discovery and quantification of a shift in temporal distribution, recording that the data in 2025 has much different properties than the data in 2024 that the validity data.
5. LSTM V-curve pattern identification which has catastrophically bad performance at 1 to 6 hour horizons yet is performing best at 12 hour horizon.
6. Analysis of operational reliability of a negative R-squared model that can be operationally valuable provided it beats persistence.
7. Physics-informed feature engineering that produces 48 features that are based on coastal hydrodynamics.

9.2 Key Findings

Models outperform persistence at operationally relevant horizons despite the difficulties with 3.5 percent at 6 hours and 14.2 percent at 12 hours. It is essential to select models that are horizontal-specific: Ensemble or Persistence when taking 1 hour, XGBoost when taking 6 hours and LSTM when taking 12 hours. The training of diversity is the most important since the performance degrades 3 to 6 times to prove that 1 year training is ineffective. There are extreme events that are difficult to predict with all the models underestimating the unprecedented January 1 st event. It has operational reliability which can withstand a degradation with 63-99 percent of predictions falling within the bounds of plus or minus 15 centimeters.

9.3 Limitations and Future Work

It has been limited to training diversity (models trained on 3 years with calm validation year) and lacks weather prediction (only makes use of observations), was only focused on a single location (predicts Packery Channel), has not measured uncertainty (only gives point predictions without confidence intervals), or adjusted to changing conditions (it is not an online learning model).

Additional work is planned in the next future working on short-term (3 to 6 months) training optimizations in the full 2021 through 2024 period, to include NOAA weather forecasts, Bayesian ensemble-estimated uncertainty quantifications, and trend / detrending features. Medium term (612 months) work Media In Media travel Station specific Full Networks Supporting online learning Hybrid physics-Machine learning models and Real-Time distribution shift detectors. Lasting (12+ months) labor encompasses validation of Transformer designs, incorporation of satellite remote weather data, multi-task learning, ensemble of ensembles development, and deployment of working system with human task checks.

9.4 Final Thoughts

This project also showed that machine learning can be valuable to coastal water level forecasting, however, it takes a strict test on really unseen data, multi-regimen training data, use of physics-informed features using domain knowledge, horizon-potential models, implemented in real decision-making, and unafraid of reporting limitations and failure modes.

The fact that significant temporal distribution shift was made is a clue to an inherent problem inherent in the past being an imprecise predictor of the future. Historical models will necessarily experience an outside case to the conditions they have been trained on. It is not to make perfect predictions but to construct models that perform well in the face of uncertainty, measure their uncertainty, as well as continue to be useful in situations they have never previously faced.

The models that have been developed here outperform persistence forecasting at operationally relevant 6 to 12 hour horizons despite the loss of performance and thus enable further improvement to occur as well as acceptable compliance with thermal performance. These models may improve the performance in 30 to 40 percent when using the high-priority enhancements mentioned above and offer credible operation support of all maritime navigation, flood warning, and coastal infrastructure protection.

References

Data Source: Texas Coastal Ocean Observation Network (TCOON). Texas A&M University. Available at: https://tcoon.tamu.edu

1. Chen, T., and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
2. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
3. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.
4. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
5. Paszke, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32.
6. Pugh, D., and Woodworth, P. (2014). Sea-level science: Understanding tides, surges, tsunamis and mean sea-level changes. Cambridge University Press.
7. Dean, R. G., and Dalrymple, R. A. (1991). Water wave mechanics for engineers and scientists. World Scientific Publishing Company.
8. Zhang, Y., et al. (2020). Real-time water level prediction in coastal areas using LSTM neural networks. Ocean Engineering, 201, 107158.
9. Makarynskyy, O. (2018). Short-term water level prediction using artificial neural networks. Marine Pollution Bulletin, 136, 296-306.

Appendix A: Complete Feature List

48 features organized by category:

1. **Tidal Features (8):** M2 sine and cosine, S2 sine and cosine, N2 sine and cosine, K1 sine and cosine
2. **Lunar Phase (2):** Lunar phase position, spring-neap indicator
3. **Wind Features (9):** U-component, V-component, and wind stress for Stations 005, 013, and 202
4. **Pressure Features (8):** Current pressure at 005 and 013, 1-hour/3-hour/6-hour changes, pressure acceleration, north-south gradient, bay gradient
5. **Temperature Features (5):** Water temperature at 005 and 013, 1-hour gradient, spatial differences
6. **Spatial Water Level (6):** North-south gradient, bay-Gulf gradient, gradient rate, spatial mean, spatial standard deviation, anomaly
7. **Historical Lags (10):** 1/6/12/24/48-hour lags, rolling standard deviations (6/12/24-hour), 24-hour mean, 1-hour rate
8. **Temporal Cyclic (4):** Hour sine/cosine, day sine/cosine

Appendix B: Hyperparameters

XGBoost:

- 1000 trees, max depth 7, learning rate 0.01
- Subsample 0.8, column sample 0.8
- Min child weight 3, gamma 0.1

Random Forest:

- 500 trees, max depth 15
- Min samples split 5, min samples leaf 2
- Max features square root (approximately 7)

LSTM:

- Input size 48, hidden size 64, 2 layers
- Dropout 0.2, sequence length 24
- Learning rate 0.001, batch size 32