University of Cape Town

Coastal Water Level Prediction: A Comparative Study of Statistical and Machine Learning Techniques for Time Series Forecasting

Author:
Jonathan Harrison

Supervisors:
Dr. Birgit Erni
Mr. Stefan Britz
Prof. John Largier

Minor dissertation submitted in partial fulfillment of the requirements for the degree of M.Sc. Advanced Analytics

at the

Department of Statistical Sciences

August 18, 2025

Abstract

Jonathan Harrison

Coastal Water Level Prediction: A Comparative Study of Statistical and Machine Learning Techniques for Time Series Forecasting

Time series analysis provides powerful tools for predicting future trends, outcomes, and events. The application of these tools to coastal water level forecasting generates insightful predictions for operational use in flood management, as well as a deeper understanding of the influencing factors. Many existing models and projects focus on long-term trends in coastal water levels particularly in terms of climate change and global warming. This project investigated the application of time series analysis with exogenous meteorological variables to the task of generating accurate short-term ($\leq$ 96 hour) forecasts of coastal water levels in a manner that is compatible with real-time monitoring for use in operational flood management. Traditional statistical methods, including regression, autoregressive integrated moving average (ARIMA), and generalised additive models, were compared alongside machine learning methods including extreme gradient boosting, support vector machines, and long short-term memory networks. Extreme gradient boosting with 24-hour of lagged input features was found to have the greatest overall test accuracy and stable predictions over the 96-hour forecast horizon. ARIMA models were the most accurate at predicting water levels in the positive stage (during high-tide). The exogenous meteorological variables contributed significantly to the models' ability to predict the water level.

Key words:
Time series, Short-term Forecasting, Regression, Machine Learning, Flood Prediction, Tidal Forecasting

Acknowledgements

Dr Birgit Erni, Prof John Largier, and Mr Stefan Britz – sincere gratitude to my supervisors for all their hard work, time and input.
Prof Sue Harrison and Mr John Harrison – special thanks to my parents for their support and encouragement.

Contents

List of Figures

List of Tables

List of Abbreviations

ACF	**A**uto**C**orrelation **F**unction
ADF	**A**ugmented **D**ickey-**F**uller
AIC	**A**kaike's **I**nformation **C**riterion
ANN	**A**rtificial **N**eural **N**etwork
ARIMA	**A**uto**R**egressive **I**ntegrated **M**oving **A**verage
ARIMAX	**A**uto**R**egressive **I**ntegrated **M**oving **A**verage with e**X**ogenous variables
ARMA	**A**uto**R**egressive **M**oving **A**verage
AP	**A**tmospheric **P**ressure
GAM	**G**eneralised **A**dditive **M**odel
GLM	**G**eneralised **L**inear **M**odel
LSTM	**L**ong **S**hort **T**erm **M**emory
LW	**L**ocal **W**ind
MAE	**M**ean **A**bsolute **E**rror
MLR	**M**ultiple **L**inear **R**egression
NF	**N**apa **F**low
OW	**O**cean **W**ind
PACF	**P**artial **A**uto**C**orrelation **F**unction
RBF	**R**adial **B**asis **F**unction
RMSE	**R**oot **M**ean **S**quared **E**rror
SD	**S**tandard **D**eviation
SVM	**S**upport **V**ector **M**achine
SVR	**S**upport **V**ector **R**egression
XGBoost	E**x**treme **G**radient **Boost**ing
XGB	E**x**treme **G**radient **B**oosting

Chapter 1

Introduction

1.1 Coastal Flooding

Many coastal areas are at constant risk of flooding – this is particularly true for low-lying coastal regions such as areas around San Francisco Bay in California, USA (Largier et al., 2023), and Table Bay and False Bay in Cape Town, South Africa (Brundrit, 2009), as well as Low Lying Islands and Coasts (LLIC) (Magnan et al., 2019). This risk is expected to increase as sea-levels rise and other climatic drivers worsen due to climate change (Reguero et al., 2015). The ability to accurately generate short-term predictions of flood risk in coastal areas is important for both the planning and operation of flood management schemes that aim to safeguard lives, property, and ecosystems.

Coastal regions are at increased risk of flooding due to the effect of storm conditions and heavy rainfall on coastal water levels. This is only being exacerbated by the global phenomenon of rising sea levels. Timely and precise predictions provide operational benefits for flood management, enabling authorities to issue early warnings, coordinate evacuations, and implement protective measures. These actions help prevent damage to infrastructure and property, reduce economic losses, and save lives. Additionally, accurate predictions offer logistical benefits, such as re-routing traffic and optimising the deployment of emergency services.

From a scientific perspective, modelling the water level could enhance understanding of coastal water systems, contributing to the knowledge base of the driving factors behind the increased water levels. This dual approach of operational and scientific benefits ensures that communities can not only respond more effectively to immediate threats but also develop strategies to mitigate future risks.

Given a dataset containing four years of water level and meteorological time series data from a station in San Francisco Bay, the aim of this project is to explore the literature; select, train, and fit a number of statistical and machine learning models capable of modelling and predicting the time series data; evaluate the performance of the models on an unseen testing subset of the data using a selection of metrics

and assess the performance of each model in terms of forecast accuracy as well as the interpretability of the model parameters.

1.2 Modelling Hydrological Systems

Maspo et al. (2020) identified two primary categories of models for modelling hydrological systems and predicting floods, the first of which entails models based on physical principles that capture the mechanistic processes which govern the overall system. These processes include, inter alia, tides, drainage mechanics, and pressure systems. In order for such models to accurately represent reality, they must be highly complex, site-specific, computationally expensive, but more importantly, they have very specific data requirements. The domain-specific knowledges, detailed data requirements and expensive computation can limit the implementation of these models, specifically for short-term forecasting application (Mosavi et al., 2018). Rahman et al. (2018) further discussed the drawbacks of physical hydrological models, citing specific data requirements and complex model calibration on a case-by-case basis and how data-driven approaches, including regression and Artificial Neural Networks (ANNs), provide alternative methods that can make use of more general, easily-available data and can be implemented by a wider range of stakeholders.

The second category specified by Maspo et al. (2020) is data-driven models, which encapsulate a broad range of methods that aim to predict outcomes based on observed patterns in historical data. While these models require a much lower level of industry- and site-expertise to construct, they do not provide the same level of understanding of the driving factors behind the predicted results. The simplest models would require only a historical time series of the response variable – i.e. the water level – but the models can be expanded to include multivariate time series of meteorological and hydrological covariates hypothesised to have a relationship with the response. Data-driven models can broadly be divided into statistical models (including both frequentist and Bayesian approaches) and machine learning techniques.

The traditional statistical methods include multiple linear and polynomial regression, autoregressive integrated moving average (ARIMA) models, and generalised additive models (GAMs). Although these models are substantially less complex and less computationally expensive than the physical-process models, they can suffer in terms of accuracy and robustness, particularly for short-term predictions (Mosavi et al., 2018). These models can provide some level of interpretability, albeit much less than the physical-process models, via the model parameter estimates. The magnitude and significance of the different input variable coefficients allow the user to understand how the different components of the model drive the resulting forecast. The forecasts include point estimates and prediction intervals which give a measure of the uncertainty in the model.

Bayesian forecasting provides an alternative to the frequentist approach to inference for these models. Under the Bayesian framework, an a priori understanding of the system is combined with observed data to generate a posterior distribution of parameter estimates. These posterior distributions can provide a more detailed evaluation of the uncertainty or variance in the forecast relative to the frequentist methods. This improved depiction of the forecast uncertainty would be very useful in terms of processing the continuous water level forecast and assessing the probability of flood events.

Machine learning techniques span a broad range of numerical methods and models that can be adapted to many problems. These methods have a number of properties that make them an attractive option for application to this study's short-term forecasting problem. The models are able to capture non-linearity in the system; are entirely informed by historical patterns with little-to-no understanding of underlying physical mechanisms; are highly flexible and easy to implement; and different model choices and tunings offer a wide range of complexity and computational costs. Models with high computational costs can often be offset by using transfer learning techniques. Possible machine learning models include ANNs, Support Vector Regression (SVR), Long Short-Term Memory networks (LSTMs), Bayesian Neural Networks (BNNs), random forests, gradient boosting, Extreme Gradient Boosting (xgBoost), and ensemble approaches.

This study will explore the literature on data-driven modelling approaches to identify a number of statistical models and machine learning methods that are capable of generating time series forecasts using autoregressive and exogenous inputs; train, fit, and tune the identified models; assess their performance on out-of-sample data; and identify the model that gives the most accurate results.

1.3 Research Aims and Data

The primary aim of this study is to develop a computationally efficient, site-specific model capable of generating accurate, short-term (96-hour) coastal water level forecasts.

To achieve this, the study uses the dataset from the Highway 37 Project (Munger et al., 2022), which includes water level measurements, tidal predictions, and a range of meteorological covariates such as atmospheric pressure, local wind, ocean wind, and river flow (used as a proxy for rainfall-induced effects). The data was recorded at Novato Mouth station in San Francisco Bay, where the water level is primarily influenced by the Bay conditions (Largier et al., 2023). This dataset is provided by Prof. John Largier, Bodega Marine Laboratory, University of California, Davis, and is publicly available online.

We fit and evaluate various statistical and machine learning models and methods. The performance of the models are compared using a predefined set of metrics with a primary emphasis on forecast accuracy on unseen data, with interpretability as a secondary consideration. The selected model should be suitable for generating accurate predictions in near real-time, while being computationally efficient enough for frequent retraining as new data becomes available. By systematically comparing forecast accuracy, this study aims to identify a robust approach for forecasting coastal water levels.

Objectives:

1. Replicate the model architectures implemented by Largier et al. (2023).
2. Fit the following statistical models:
    - Multiple Linear and Polynomial Regression models.
    - Autoregressive Integrated Moving Average models.
    - Generalised Additive Models.
3. Train and tune the following machine learning models:
    - Extreme Gradient Boosting models.
    - Support Vector Regression models.
    - Long Short Term Memory models.
4. Assess forecasting performance of models and methods identified above.
    - Investigate and select appropriate metrics to evaluate the accuracy, reliability, and interpretability of the models.
    - Assess model performance by generating forecasts using a previously unseen test subset of the data.
    - Compare the accuracy and reliability of the model throughout the 96-hour forecast horizon.
    - Assess forecast accuracy across different prediction horizons within the 96-hour forecast horizon.
    - Evaluate accuracy at varying tidal states.
    - Assess the ability to interpret model output to understand the relationship between covariates and response.
5. Identify the model or method that is best suited to achieve the above aim in an operational setting.

1.4 Dissertation Layout

Chapter 2 starts by reviewing the existing literature on water level modelling and prediction, and time series analysis, including traditional statistical approaches, Bayesian methods and machine learning techniques. Chapter 3 introduces the dataset of interest, reproduces and discusses the models created by Largier et al. (2023) for generating short-term water level forecasts, and proposes alternative approaches to solving the problem. We then give the methodologies for the statistical and machine learning models that will be applied to solve the forecasting problem in Chapter 4, and analyse the results of these models in Chapter 5, where we discuss the suitability, performance, and level of interpretation available for each model created. Finally, Chapter 6 provides a conclusion of the study's key findings, as well as future work and recommendations.

Chapter 2

Literature Review

With the aim to identify suitable models for the short-term coastal water level forecasting problem, this chapter explores the existing literature on statistical and machine-learning methods for time series analysis, water level forecasting, the use of exogenous variables, and trade-offs between different approaches and model architectures.

2.1 Coastal Water Level Forecasting

Huang et al. (2003) implemented a 3-layer Artificial Neural Network (ANN), 25 units per layer, to generate forecasts of the coastal inlet water level in Long Island South Shore based on inputs from remote water level stations. The model used 4 hours of lagged water level data as input, providing a autoregressive property to the model. Han et al. (2008) undertake a project with a similar objective - to predict water level at a specific site based on input data gathered at remote stations. The study implemented a three-layer ANN and a neural network ensemble that combined the output of a group of member networks using a trimmed average algorithm. The trained models were able to provide accurate and robust long-term predictions of water level given a short period of lagged input. Lee et al. (2014) used artificial neural networks and generalised regression neural networks to forecast coastal high and low water levels. Despite good model fit for training data, the models performed poorly for forecasts horizons of both one month and one year, with the one-year forecast suffering from much greater errors.

2.2 Statistical Methods

Multiple Linear Regression (MLR) models leverage multiple predictor variables to establish linear relationships with a dependent variable, but are not able to capture complex, non-linear dynamics (Damle, 2005). Multiple polynomial regression extends these models by introducing higher-degree terms that can be used to model non-linear relationships, at the cost of increased model complexity. Largier et al. (2023) developed a system for site-specific forecast of water levels at three stations (Novato

Creek, Petaluma River, Rowland Bridge) along the State Highway 37 in California, USA. This system predicted the coastal tide, calculated a tidal residual, and then modelled the residual to determine the effect of a number of meteorological and hydrological factors on water level. The model combined multiple regression with an autoregressive moving average forecast error correction. After a model fitting process, the final model used a cubic polynomial with four covariates, resulting in a model with 34 regression coefficients.

GAMs are an extension of the GLM class that model the response variable as a sum of arbitrary smooth functions of the predictors, allowing the capture of non-linear relationships between the response and predictors (Hastie et al., 1986). Dubos et al. (2022) investigated the use of Generalised Additive Models (GAMs) applied to the task of generating short-term forecasts of peak river flow intensity with the intention to predict spring flooding in boreal river systems. Dubos et al. (2022) compared the performance of Generalised Additive Models (GAMs), Generalised Linear Models (GLMs) and a deterministic hydrological model applied to that task. Despite the geographical differences between those boreal river systems and the Mediterranean-climate, coastal water systems in this study, the framework of the problem is very similar – i.e. aim to predict a continuous measure using a simple dataset of hydrological and meteorological variables. A key component of fitting a GAM model is the choice of smoothing function. Hastie et al. (1986) used a running lines smoother, but suggested that alternative functions such as a kernel or spline smoother would also be viable. Dubos et al. (2022) used penalised cubic regression splines to reduce risk of overfitting by limiting the degrees of freedom in the smoothing function.

Rahman et al. (2018) used GAMs to model flood frequency quantiles, as the smooth functions allow the model to capture the non-linear relationship between the covariates (i.e. rainfall) and the response. They opted for thin-plate regression splines, noting the computational efficiency as well as the advantages of not needing to specify the knot locations, and the optimal approximation of smoothness. Dominici et al. (2002) utilised GAMs for modelling non-linear time series data while investigating the impact of air pollution on health. They explored using local regression smoothers and natural cubic splines with equi-spaced knots and found that the natural cubic splines performed better. They also caution against the use of the default settings when implementing GAMs in S-PLUS.

Yurttas (2024) used various autoregressive moving-average (ARMA) models to generate forecasts for water levels in the Twente Canal in the Netherlands. This project explored a number of variations to the ARMA model including integration, seasonality, and exogenous variables. The integration extension allows the model to handle non-stationary data by differencing the time series (Tyagi et al., 2023). Exogenous variables are variables that exert influence on the response variable, but are not influenced by the response variable in return (Box et al., 2015). The exogenous variables allow the model to capture additional information from these influential inputs. The

seasonal variation of the model introduces seasonal autoregression, seasonal differencing or integration, and seasonal moving averages (Banas et al., 2021). Yurttas (2024) concluded that although the ARIMA models performed poorly for long-term forecasts since the predictions revert to the training sample mean, the models were able to generate short-term forecasts with a high level of accuracy.

Tyagi et al. (2023) compared the performance of ARIMA and ARIMAX models, among other models, applied to the problem of forecasting sugarcane production in India and concluded that the ARIMA model (without exogenous variables) was the most suitable model with the best model fit, as measured by the Akaike Information Criterion (AIC).

2.3 Machine Learning Methods

Flooding, tidal water levels and other geophysical and hydrological phenomena are the result of complex systems with many affecting factors. Damle (2005) discussed how classical statistical methods such as linear regression and ARIMA models are unable to capture the complexities of a river water system and their performance suffers due to the difficulties in satisfying the requisite model assumptions – such as normality, independence, and stationarity – going on to suggest that non-linear processes such as neural networks are better able to generate accurate short-term forecasts. Damle (2005) further investigated the use of time series data mining, a methodology based on a variant of time delayed embedding, applied to predicting flood events – which is beyond the scope of this study.

Tsakiri et al. (2018) investigated MLR and ANN models applied to the problem of flood prediction in an urban river system. The comparison indicated that the ANN model improved the accuracy of forecasting flood events and explained a greater proportion of the time series variance relative to the MLR model, but the MLR model provided scope for interpretation of the results.

Lindemann et al. (2021) stated that Long Short-Term Memory networks (LSTMs) are well-suited to the task of time series prediction. Lindemann et al. (2021) further suggested that hierarchical and attention-based LSTM architectures are well-suited to handle multidimensional data; Grid and cross-modal LSTM architectures are suited to predicting multiple quantities with high precision; and partially conditioned Seq2Seq LSTMs are able to model long- and short-term dependency.

Ribeiro et al. (2019) explored various bagging, boosting, and stacking ensemble regression techniques for time series prediction with the application to predicting agricultural commodities prices. They suggested that Extreme Gradient Boosting (XGBoost), gradient boosting machines (GBM) and random forests (RF) are the ensemble techniques best suited to time series forecasting.

Fang et al. (2022) undertook a time series study that compared the performance of an ARIMA model against an XGBoost model in predicting the occurrence of COVID-19 cases in the US. The XGBoost model incorporated temporal aspects by including a set of seven daily lagged terms (of the response variable). The model also included a one-hot encoding of the day of the week as well as a time variable to capture the trend over time. Cross-validation was used to select optimal values for the hyperparameters within preset bounds. The models were compared in terms of accuracy, which was quantified by root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) metrics. Fang et al. (2022) concluded that the XGBoost model outperforms the ARIMA model – both in terms of the in-sample model fit as well as the out-of-sample test forecast – across all three accuracy metrics.

Support Vector Regression (SVR), an adaptation of the Support Vector Machine (SVM) which was originally designed for classification problems, provides an efficient, non-linear and data-driven approach to predict a continuous outcome. Appropriate feature architecture can adapt the model to time series prediction by designing lagged features to introduce a temporal component to the model. Sapankevych et al. (2009) compiled a survey summarising research applications of SVM methods to time series forecasting across a range fields with a focus on the numerical accuracy of the prediction while taking into consideration model tuning and computational cost. Their findings state that primary benefits of SVR for time series forecasting include the model's data-driven approach, which means there is little need for a priori input, the flexibility afforded to the user in its design and implementation, and foremost the ability to handle nonlinearity via the kernel function trick. This allows the model to enjoy the benefits of transforming the data to a higher-dimensional space where linear techniques can be employed to find non-linear solutions in the original space without having to explicitly compute the transformation.

Sapankevych et al. (2009) note the limitations and drawbacks of SVR, stating that the tuning process is very important, as there are limited heuristics for choosing hyperparameters, the choice of kernel function tends to be fairly arbitrary as there is no method for optimal selection, and there is no standard method for quantifying the uncertainty of the prediction.

Trafalis et al. (2000) and Tay et al. (2001) evaluated the performance of SVR models against feed-forward neural networks with backpropagation for the tasks of predicting stock prices and financial indices. Both papers found that SVR was able to outperform the neural networks. Tay et al. (2001) suggest that SVR outperformed neural networks, as it was better able to generalize to unseen data, while neural networks were more likely to suffer from overfitting. Sansom et al. (2002) also compared SVR with a feed-forward neural network applied to the task of forecasting electricity prices. The models performed similarly in terms of prediction accuracy, but the SVR model enjoyed lower computational cost of training.

Hochreiter et al. (1997) proposed an adaptation of the recurrent neural network (RNN) called long short-term memory (LSTM) that can handle complex, long-term lag problems that were previously beyond the scope of RNNs. Ogunmolu et al. (2016) compared the performance of three deep neural network architectures, namely multi-layer networks, simple recurrent networks and long short-term memory variants, used for modelling non-linear systems.

Cho et al. (2022) compared the performance of three different recurrent neural network architectures applied to the task of predicting water level in a river system. The first architecture used LSTM hidden layers, the second architecture uses gated recurrent unit (GRU) layers, and the third architecture uses a combination of the two. LSTM networks are able to maintain dependencies by holding information in long-term states – a 'forget gate' determines how much information is retained, while input and output gates control the flow of information in and out of memory cells.

The GRU architecture is another adaptation of the RNN that aims to maintain long-term dependencies similar to the LSTM network, but with lower computational needs and fewer parameters – the forget, input, and output gates are replaced by an update gate and a reset gate that control the flow of information over time. Cho et al. (2022) found that the LSTM models outperformed the GRU models according to the test mean square error (MSE), but the combined LSTM-GRU models performed the best with the lowest test MSE and the smallest maximum prediction error.

2.4 Bayesian Methods

Harrison et al. (1971) described an Bayesian approach to short-term forecasting. In this approach, each data point is used to calculate the posterior probability at that moment in time. This allows the model to generate both individual point forecasts as well as the distribution of parameter values, giving a thorough understanding of the uncertainty of the prediction.

Brooks et al. (2008) explored a Bayesian Monte Carlo Markov Chain (MCMC)-based approach that allowed them to estimate the probability of the forecast exceeding some threshold. This kind of approach could be well suited to the problem of forecasting water level, as in addition to the forecast estimate, it would provide the probability of the water level exceeding some flooding threshold, giving an explicit measure of flood risk.

Zhu et al. (2017) presented a novel end-to-end Bayesian deep-learning model that provides time series prediction capabilities with a level of understanding of the prediction uncertainty far beyond standard machine learning methods. These Bayesian Neural Networks (BNNs) provide an uncertainty estimation that decomposes the prediction uncertainty into model uncertainty (uncertainty in model parameters), inherent noise

(irreducible uncertainty in the data generating process), and model misspecification (uncertainty in the relationship between sample and population) (Zhu et al., 2017).

Although Bayesian methods have a lot to offer when approaching the problem of coastal water level forecasting and the associated flood prediction, no Bayesian models will be implemented or further discussed in this study due to time and resource constraints.

Chapter 3

Data and Model Replication

The accuracy and reliability of predictive models are highly dependent on the quality and preparation of the data used for training and evaluation. For this study, the dataset provided by the Highway 37 Project (Munger et al., 2022) serves as the foundation for developing and testing a number of statistical and machine learning models for forecasting coastal water level. To ensure robust model evaluation, the dataset is partitioned into training and test subsets. Using the same dataset across all models provides a consistent baseline for performance comparison. This chapter will replicate and examine the regression models and approach of Largier et al. (2023) in detail.

3.1 Data Overview & Preparation

Largier et al. (2023) developed a data-driven, empirical system for site-specific forecasting of coastal water level in low-lying coastal areas in the North of San Francisco Bay, California for a National Center for Sustainable Transportation (NCST) research project. For this task, they built a dataset from existing data sources, containing water level data, tidal predictions, and a number of meteorological factors. This dataset (Munger et al., 2022) is used in this project to train, fit, and test the candidate models. The dataset contains a collation of publicly-available data sourced from the National Data Buoy Center (NDBC), the National Weather Service, and the National Water Prediction Service divisions of the National Ocean and Atmospheric Administration (NOAA) agency; Water Data for the Nation hosted by the United States Geological Survey (USGS); Sonoma Water, a water-monitoring initiative in Sonoma County, California; Marin County Flood Control and Water Conservation District (MCFCWCD); and OpenWeatherMap, an open-source weather data service. The dataset covers the period from 2019-01-01 to 2022-09-27.

The water level data was measured at three stations around Highway 37 in San Francisco Bay by the MCFCWCD. These stations are located at the mouth of Novato Creek, at Rowland Bridge on Novato Creek, and at Sonoma Horse Park on the Petaluma river. For this study, we will use data from the Novato Creek mouth station, shown in Figure 3.1, as it is the most complete dataset. In order to obtain the

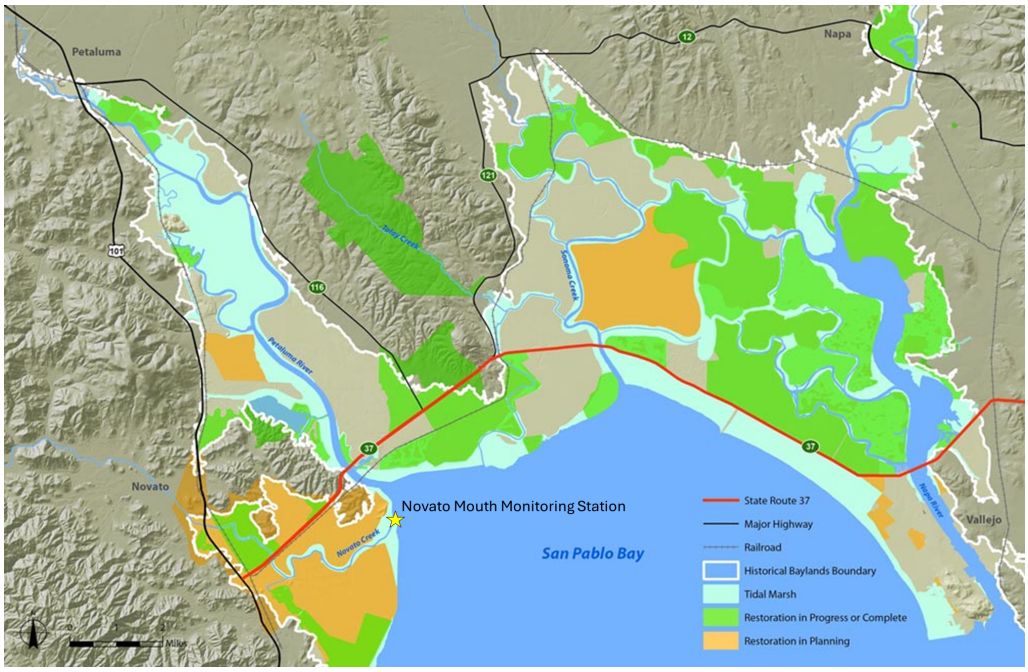

FIGURE 3.1: Map of San Pablo Bay and North Bay tidal wetlands (Trust, 2021) in the San Francisco Bay system. Highway 37 is shown in red. The Novato Mouth station run by Marin County is denoted by the yellow star.

residual water level (RWL), Munger et al. (2022) processed the data as follows: the measured water level, also referred to as the stage level, was centred (mean zero) and small gaps in the hourly time series were imputed using linear interpolation. The predicted tide (TP) was obtained using unified tidal analysis and calculated using a publicly available Python distribution of the Matlab package UTide. The RWL was then calculated as the difference of the observed water level and the tidal prediction. It is measured in metres (m) and captures the movements of the water level due to non-tidal forcing. It should be noted that the tidal analysis excluded low-tide data, so the residuals were only calculated at high tide – roughly every 12 hours. The 12-hourly time series was then smoothed using a three-point window and interpolated to hourly observations (Largier et al., 2023).

Largier et al. (2023) investigated a number of potential predictors including precipitation, atmospheric pressure, local onshore wind, ocean onshore wind, sea-surface temperature, river flow and river stage (water level). A subset of predictors was selected for model training based on a correlation analysis as well as considering the ease of access and reliability of 3–5 day forecasts for the predictors. The chosen subset was comprised of Napa River flow (NF), measured in cubic feet per second (cfs); atmospheric pressure (AP) at the Bay mouth, measured in millibar (mBar); onshore wind over the ocean, this will be referred to as ocean wind (OW), measured in metres per second (m/s) from the South-Southwest (SSW); and local onshore wind, referred to as local wind (LW), measured in metres per second (m/s) from the Southeast (SE). The summary statistics are given in Table 3.1.

	Min	1st Qu.	Median	Mean	3rd Qu.	Max
Atm. Pres.	994.6	1014.2	1016.9	1017.3	1020.4	1036.9
Local Wind	-10.57	-2.19	-0.42	-0.55	1.37	13.55
Napa Flow	0.00	0.00	4.3	118.3	31.8	16600.0
Ocean Wind	-13.40	-3.68	-1.68	-1.26	0.76	15.38
Predicted Tide	-1.018	-0.336	-0.006	0.016	0.365	1.177
Stage Level	-0.940	-0.409	-0.011	0.004	0.381	1.335
Residual Level	-0.260	-0.044	0.002	0.006	0.049	0.488

TABLE 3.1: Summary statistics for the variables, units are specified above.

The Napa River flow measures inflow to the bay and acts as a proxy for rainfall. The atmospheric pressure is negatively correlated with the water level as a barometer effect causes the water level to rise for any drop in atmospheric pressure. The SE wind blows across the bay, pushing surface water towards Novato Creek and piling up water along the shore of the North Bay. Largier et al. (2023) also noted that the SSW wind may be a proxy for other factors including Ekman upwelling – a component of wind-driven ocean current – and geostrophic currents.

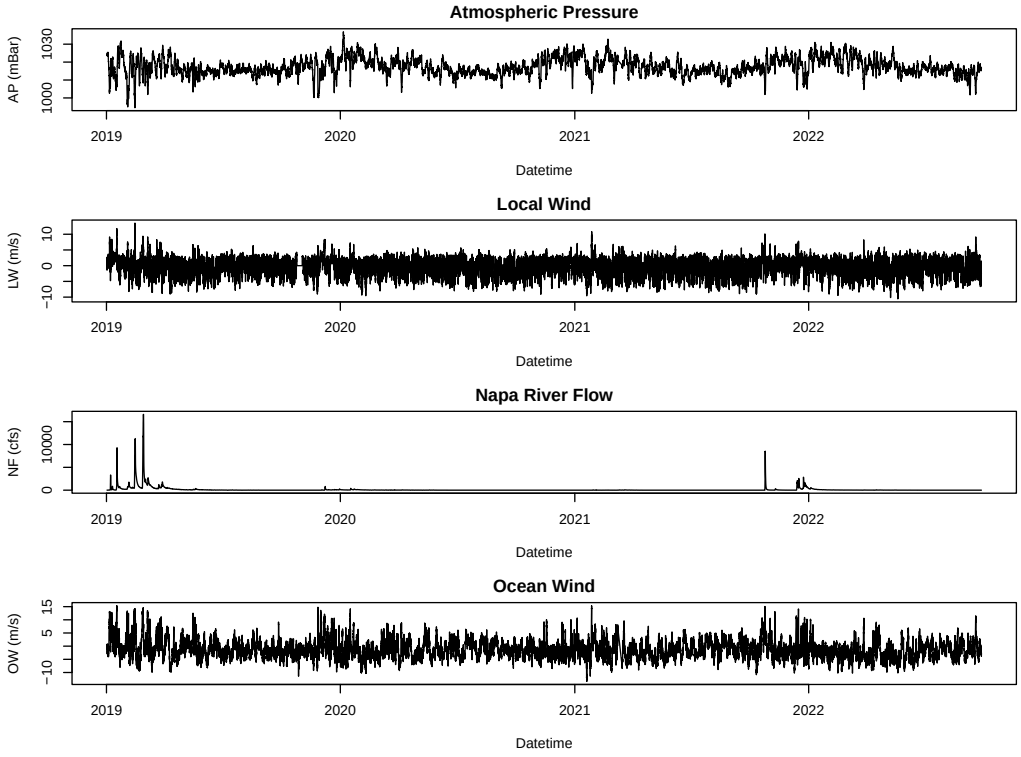

FIGURE 3.2: Time series plots of the meteorological predictor variables.

We split the data into training and testing subsets. The training set runs from 2019-01-01 20:00:00 to 2022-06-17 15:00:00 and contains 30 308 observations, just shy of

42 months (1263 days) of hourly observations. The testing set runs from 2022-06-17 16:00:00 to 2022-09-27 15:00:00 and contains 2448 observations, equivalent to 102 days. The time series for the meteorological predictor variables are visualised in Figure 3.2, and the water level variables are given in Figure 3.3. Note, we assume that values for the covariates will be available for the duration of any forecast (up to 96-hours) – these would be gathered from weather forecast data, which would provide much more accurate values than attempting to forecast these covariates in this study.

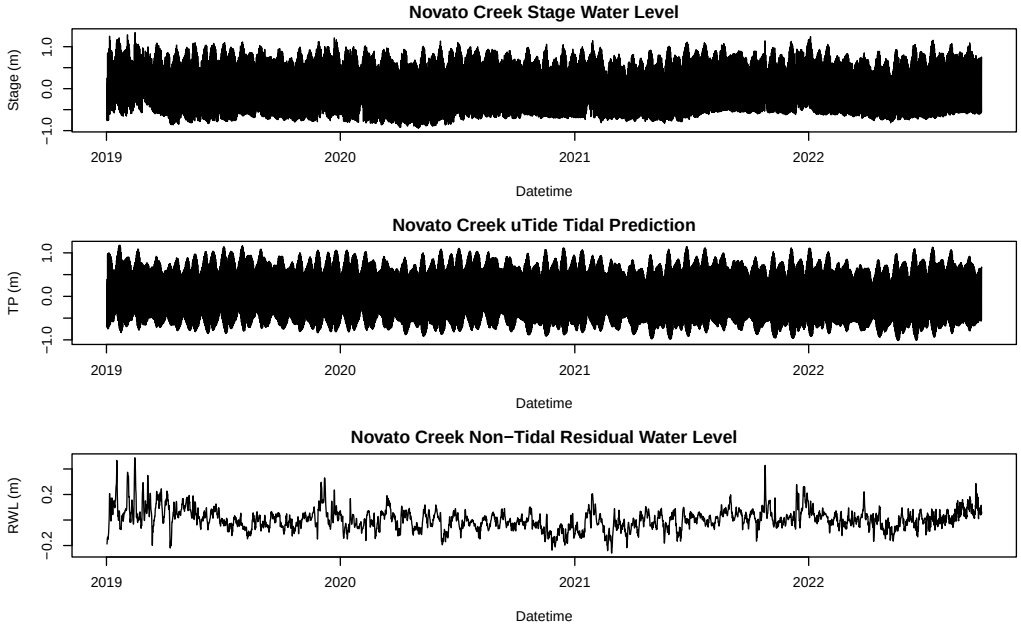

FIGURE 3.3: Time series plots of the stage, uTide tidal prediction, and residual water levels.

3.2 Replication of NCST Models

The first three regression models replicate those implemented by Largier et al., 2023 – multiple polynomial regression models of linear, quadratic, and cubic order. Increasing the order of the polynomial terms rapidly increases the complexity of the model and the number of coefficients that must be estimated from the data. As the number of model coefficients increases, it also becomes more difficult to interpret the effects of individual predictors on the response. The number of coefficients increases quickly with the order of the polynomial, from 5 in the linear regression model, with four predictors and an intercept term, to 15 in the full quadratic regression model, to 35 in the full cubic regression model. For brevity's sake, the full model form is only written explicitly for the linear model in Eq. 3.1.

$$\hat{y}_t^{(\mathrm{RWL})} = \beta_0 + \beta_1 x_t^{(\mathrm{AP})} + \beta_2 x_t^{(\mathrm{LW})} + \beta_3 x_t^{(\mathrm{NF})} + \beta_4 x_t^{(\mathrm{OW})}, \tag{3.1}$$

where $\hat{y}_t^{(\text{RWL})}$ denotes fitted RWL for the t^{th} observation; x_t^{AP}, x_t^{LW}, x_t^{NF}, x_t^{OW} refer to atmospheric pressure, local wind, Napa river flow, and ocean wind for the t^{th} observation respectively.

Largier et al. (2023) suggested implementing an error correction to handle long-term, large-scale factors such as changes to the climate and currents, as well as phenomena like the El Niño and La Niña cycle patterns. The error correction is based on the assumption that model errors close together in time will be similar, as the model error varies slowly over time – Largier et al. (2023) showed that the 12-hour rolling mean error is not substantially different from the seven-day rolling mean error. They further suggested that water level prediction can be improved in terms of short-term forecast accuracy by adjusting the predicted value by the average error of predictions for the last 12 hours and state that this correction can improve forecast accuracy up to three days into the future, after which the error correction degrades the accuracy of the predictions.

The three linear regression models are trained using the training subset of the data. The out-of-sample performance is then assessed by generating multiple 96-hour forecasts throughout the testing dataset – the 96-hour forecast horizon is consistent with Largier et al. (2023)'s forecast period of four days. For each forecast, the model generates hourly predictions of RWL, $\hat{y}_{t+1}^{(\text{RWL})}, \ldots, \hat{y}_{t+96}^{(\text{RWL})}$. An error correction (EC) strategy is applied – for each 96-hour forecast starting at time t, the predicted values are adjusted by the average error, $\bar{e}_t$, for a window of 12 hours directly prior to the time of forecast, t, such that $\tilde{y}_{t+1}^{(\text{RWL})} = \hat{y}_{t+1}^{(\text{RWL})} + \bar{e}_t$; where $\bar{e}_t = \frac{1}{12}\sum_{k=0}^{11}\left[y_{t-k}^{(\text{RWL})} - \hat{y}_{t-k}^{(\text{RWL})}\right]$.

The performance of each of these forecasts, both with and without the error correction, is measured using the root mean squared error (RMSE) and the mean absolute error (MAE) metrics. These metrics are then aggregated over all of the testing forecasts made to give mean RMSE and MAE, along side the range of the residual errors scores, for each model as shown in Table 3.2. On average, the error correction is able to improve the accuracy of the regression models ability to forecast RWL in terms of RMSE and MAE, however the error correction worsens the magnitude of error of the most inaccurate predictions increasing the range of errors in both directions.

TABLE 3.2: Residual Water Level Predictions: Training RMSE and aggregate performance metrics for the out-of-sample test predictions of RWL for regression models including EC approaches (units in cm).

Model	**Train RMSE**	**RMSE**	**MAE**	**Min Error**	**Max Error**
Linear	5.8	4.5	3.8	-14.8	17.7
Quadratic	5.4	4.7	4.0	-13.9	18.2
Cubic	5.3	4.7	4.0	-13.6	18.4
Linear EC	5.8	4.0	3.4	-17.6	22.0
Quad. EC	5.4	4.1	3.4	-17.8	24.1
Cubic EC	5.3	4.1	3.4	-17.4	23.1

In order to move from a prediction of RWL to a forecast of the stage water level, as performed by Largier et al. (2023), the RWL predictions are combined with the corresponding tidal prediction to give a prediction of the stage water level, such that $\hat{y}_i^{(\text{stage})} = \hat{y}_i^{(\text{RWL})} + x_i^{(\text{TP})}$, and for the EC models, $\tilde{y}_i^{(\text{stage})} = \tilde{y}_i^{(\text{RWL})} + x_i^{(\text{TP})}$. This process adds variance and uncertainty to the forecast water level, the RMSE and MAE increase for the stage forecast, relative to the RWL forecast. As shown in Table 3.3, there is little change in accuracy (RMSE and MAE) between either the models of different order or the EC models.

TABLE 3.3: Stage Water Level Predictions: aggregate RMSE, MAE, and range of errors across all test (out of sample) forecasts for regression models including EC approaches (units = cm).

Model	**RMSE**	**MAE**	**Min Error**	**Max Error**
Linear	9.0	6.8	-41.2	42.0
Quadratic	9.1	6.9	-40.1	41.6
Cubic	9.1	6.9	-39.6	41.7
Linear EC	9.1	6.8	-39.1	48.0
Quad. EC	9.1	6.9	-38.5	47.9
Cubic EC	9.1	6.8	-38.3	47.5

Table 3.4 shows the average RMSE across all test forecasts of stage water level at different forecast horizons. For the unadjusted forecasts, the forecast accuracy appears to improve as the horizon increases. This is unintuitive as one might expect better accuracy closer to the present, however, these simple regression models without adjustment do not have temporal dependence. It just so happens that the models perform worse on the data at the end of the test dataset than on the data for the first few hours of the test dataset – this means that the aggregate performance of the short-horizon forecasts includes more tests on this favourable data, while the longer-horizon forecasts include more of the unfavourable data at the tail of the dataset, biasing the metrics.

Table 3.4 is still useful to compare the difference in performance due to the error correction at different forecast horizons. One can see that for very short forecast horizons (1-12 hours), the error correction improves the forecast accuracy for all three models. However, for a 48-hour or greater forecast horizon, the error correction diminishes the forecast accuracy.

Largier et al. (2023) preferred the approach of explicitly separating the tidal and non-tidal influences of the water level. The argument for doing so is that tide accounts for the greatest contribution to water level and once it has been extracted, it is easier to model and identify other non-tidal factors affecting the residual signal. Largier et al. (2023) noted that at the Novato Mouth observation site, the uTide tidal prediction performed poorly at low-tide due to friction-dominated drainage distorting the harmonic analysis, and as a result, the tidal residual water level had large errors at low-tide. Therefore, in the original data processing, the residuals were only

TABLE 3.4: Test Performance measured by RMSE for regression models across different forecasting horizons, measured at 1, 12, 24, 48, 72, and 96 hours. Reported as centimetres with 1 decimal.

Model	1-hour	12-hour	24-hour	48-hour	72-hour	96-hour
Linear	9.5	9.4	9.4	9.2	9.1	9.1
Quadratic	9.6	9.4	9.5	9.3	9.2	9.2
Cubic	9.6	9.4	9.5	9.3	9.2	9.2
Linear EC	9.4	9.3	9.5	9.4	9.4	9.4
Quad. EC	9.4	9.3	9.5	9.4	9.4	9.4
Cubic EC	9.4	9.2	9.5	9.4	9.4	9.4

calculated at high-tide (roughly twice per day) and hourly observations in-between had to be estimated or interpolated and smoothed, a process which adds uncertainty and variance to the system.

As shown in Figure 3.4, the relationship between the water level stage and the predicted tide appears to be nonlinear – this means that if RWL is calculated as the difference between stage and predicted tide and modelled as a function of the other covariates only (excluding tide), the predictions for water level ignore this relationship and the accuracy will suffer for low-tide observations.

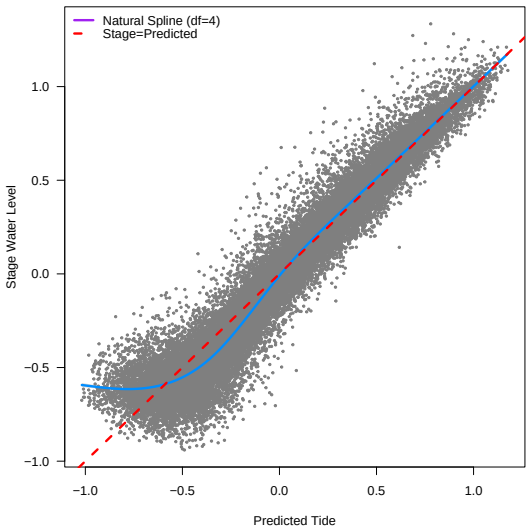

FIGURE 3.4: Scatterplot of observed water level stage against the predicted tide shows a nonlinear relationship between the two.

As seen in the QQ-plots shown in Figure 3.5, the model errors deviate from normality at the tails. The deviation at the lower tail decreases as the model complexity increases, but so does the deviation at the upper tail.

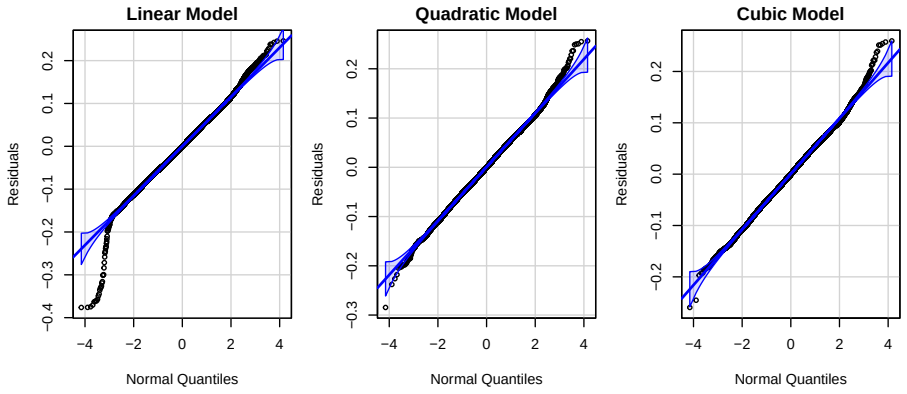

FIGURE 3.5: QQ plots showing residuals deviating from Normality for the regression models.

Moving forwards, the new models implemented will predict the stage water level directly as a function of the chosen covariates as well as the UTide tidal prediction (TP). This will allow the models more flexibility while incorporating the tidal information and has the added benefit that results will be directly interpretable, without requiring transformation back to stage-scale.

Chapter 4

Methods

This chapter introduces a naive baseline model and details the implementation of the statistical models – Multiple Polynomial Regression, Autoregressive Integrated Moving Average (ARIMA) models, Generalised Additive Models (GAMs) – and the machine learning methods – Extreme Gradient Boosting (XGBoost), Support Vector Regression (SVR), and Long Short Term Memory (LSTM) networks. Each section will specify the process of fitting and training these models, the choice of model parameters, and hyperparameter tuning. We will then describe the forecasting strategy for assessing the out-of-sample test performance of each model, the choice of metrics for evaluating accuracy, and methods for assessing variable importance and partial dependence. All the coding and model implementation is performed using `R` in RStudio.

4.1 Naive Model

The baseline model for comparison will be a naive model that simply predicts the stage water level as equal to the tidal prediction (TP) as generated by the UTide routine (see Section 3.1), which is available as one of the predictor variables. This comparison will allow one to determine if models are able to make use of the information in the additional meteorological covariates to capture the effect of the non-tidal forcing of the water level.

4.2 Statistical Models

4.2.1 Multiple Linear and Polynomial Regression Models

Linear regression models capture the relationship between response and predictor variables by fitting a linear equation to the observed data. In order for the regression coefficients to be as intuitive as possible, the atmospheric pressure (AP) and Napa river flow (NF) predictors are rescaled to $[0, 1]$, while the local wind (LW) and ocean wind (OW) variables are rescaled to $[-1, 1]$, as the sign denotes the inverse direction of the wind. Ordinary Least Squares is the standard method for estimating the parameters of a linear regression model by minimising the sum of the squared errors

between the observed and fitted values. Multiple Linear Regression (MLR) models are an adaptation of the classic linear regression models that allow for the response to be modelled in terms of multiple predictor variables. Polynomial regression is a further adaptation that allows the model to capture non-linear relationships between the predictors and the response through the addition of polynomial terms, up to order d. For example, a multiple polynomial regression of order $d = 3$ with two predictor variables and a full complement of terms, including the interaction terms, would take the form shown in Equation (4.1), where β_j is the j^{th} model coefficient, y_i is the response for the i^{th} observation, $x_{i,1}$ and $x_{i,2}$ are the respective predictor variables for the i^{th} observation, and the ϵ_i are the random errors, which are assumed to be independent and identically Gaussian-distributed for all i.

$$\begin{aligned} y_i =& \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,1}^2 + \beta_4 x_{i,1} x_{i,2} + \beta_5 x_{i,2}^2 \\ &+ \beta_6 x_{i,1}^3 + \beta_7 x_{i,1}^2 x_{i,2} + \beta_8 x_{i,1} x_{i,2}^2 + \beta_9 x_{i,2}^3 + \epsilon_i \end{aligned} \tag{4.1}$$

The full linear, quadratic, and cubic models for the stage water level will have six, 21, and 55 coefficients respectively. However, not all of these higher-order polynomial terms contribute to the model in a meaningful way. We apply step-wise variable selection to remove unnecessary terms from the full model using the `stepAIC` function from the `MASS` library (Ripley et al., 2023). The algorithm removes terms that do not contribute to improving the model fit, and aims to minimise the model's Akaike's Information Criterion (AIC) value. All the terms of the linear and quadratic models are deemed relevant to keep, while 9 terms are stripped from the cubic model, resulting in a total of 46 terms. The model forms chosen by step-wise selection are given by equations A.1 – A.3 respectively.

To capture regular cyclical patterns in the water level due to lunar and seasonal cycles, we add two pairs of orthogonal sine and cosine terms to the linear and cubic polynomial models. The first pair has a period of 27.3 days, which corresponds to the duration of the Moon's sidereal orbit around the Earth, and the second pair has a period of 365.24219 days, corresponding to the duration of the Earth's orbit around the Sun. These terms allow the model to capture the effects of the gravitational forces and seasonal patterns on the water level outside of the effects captured by the predicted tide variable used to calculate the residual water level. The trigonometric functions provide a smooth curve to include seasonality in the model, compared to using indicator variables to denote seasons in a discontinuous manner.

4.2.2 ARIMA Models

The regression models in the previous section assume that the observations are independent at each point in time. Seasonal terms and error corrections are used to add some level of time series structure to the models.

The next set of models use an ARIMA with exogenous variables (ARIMAX) framework. The ARIMA components give the models capacity to handle the temporal dependence between sequential observations, while the exogenous variables allow for the information contained in the predictor variables at each time point to be incorporated. The parameters for the autoregressive (AR), integrated, and moving average (MA) components (p, d, q) determine the number of lagged observations and terms included in the model – larger parameter values generate bulkier, more complex models. The choice of parameter values is informed by testing the time series of interest for stationarity and autocorrelation.

The Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test is a statistical test that determines if a time series is stationary. If the time series is non-stationary, it needs to be differenced to achieve stationarity. In the ARIMAX framework, this is handled by the integration component, where parameter d controls the number of differences applied. The KPSS test, implemented using the `tseries` library (Trapletti et al., 2023) and applied to the stage water level time series, returns a test statistic of 6.81 and a p-value of 0.01 – indicating the stage is non-stationary. We take the difference of the series and apply the KPSS test again. This test returns a test statistic of 0.0002 and a p-value > 0.1 – there is insignificant evidence to reject the null hypothesis of stationarity for the differenced series. This suggests that $d = 1$ is appropriate.

The Ljung-Box test tests for autocorrelation in the time series by assessing if the observations are independently distributed. The Ljung-Box test, implemented using the `tseries` library (Trapletti et al., 2023) and applied to the stage water level time series, returns a test statistic of 24053 and a p-value $<< 0.01$ – indicating that there is significant autocorrelation in the stage series. The autocorrelation of the time series is visualised in Figure 4.1 by the autocorrelation (ACF) and partial autocorrelation (PACF) functions. The model parameters are chosen using the `auto.arima` function from the `forecast` library (Hyndman et al., 2008) to conduct an algorithmic search of the parameter space and to select parameter values that minimise the AIC value of the model. This search suggests that the choice of values $p = 5$, $d = 1$, and $q = 1$ is suitable. In order to improve numeric stability and reduce convergence issues during model fitting, the meteorological covariates were re-scaled between 0 and 1.

The first ARIMAX model is fit with the parameterisation $(5, 1, 1)$, according to the AIC-minimising search. The form for this model is given by Equation (4.2) which can be simplified to take the form given by Equation (4.3).

$$\begin{aligned} y'_t = &\alpha + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + \phi_3 y'_{t-3} + \phi_4 y'_{t-4} + \phi_5 y'_{t-5} \\ &+ \beta_1 \mathrm{AP}_t + \beta_2 \mathrm{LW}_t + \beta_3 \mathrm{NF}_t + \beta_4 \mathrm{OW}_t + \beta_5 \mathrm{TP}_t \\ &+ \epsilon_t + \theta_1 \epsilon_{t-1} \end{aligned} \tag{4.2}$$

$$\begin{aligned} y_t = &\alpha + (\phi_1 + 1) y_{t-1} + (\phi_2 - \phi_1) y_{t-2} + (\phi_3 - \phi_2) y_{t-3} \\ &+ (\phi_4 - \phi_3) y_{t-4} + (\phi_5 - \phi_4) y_{t-5} - \phi_5 y_{t-6} \\ &+ \beta_1 \mathrm{AP}_t + \beta_2 \mathrm{LW}_t + \beta_3 \mathrm{NF}_t + \beta_4 \mathrm{OW}_t + \beta_5 \mathrm{TP}_t \\ &+ \epsilon_t + \theta_1 \epsilon_{t-1}; \end{aligned} \tag{4.3}$$

where y'_t denotes $y_t - y_{t-1}$, ϕ_i are the autoregressive coefficients; $i = 1, \ldots p$; θ_j are the moving average coefficients, $j = 1, \ldots, q$; β_k are the exogenous coefficients, $k = 1, \ldots, 5$; ϵ are the error terms; and α is a constant term.

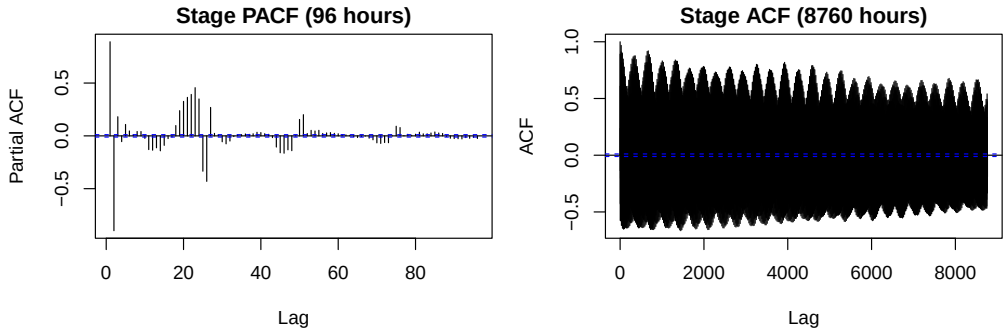

FIGURE 4.1: ACF and PACF plots for Stage water level.

We extend the ARIMAX model, adding seasonality to create a seasonal ARIMAX (SARIMAX) model. We need to identify the seasonal parameters for the AR (P), integrated (D), and MA (Q) components. We extract the 24-hour seasonal component using additive classical decomposition and apply the KPSS test for stationarity to it – this returns a test statistic of 0.00017 and a corresponding p-value > 0.1, indicating that the 24-hour seasonal component is stationary and $D = 0$ is suitable. The PACF plot in Figure 4.1 shows spikes of partial autocorrelation around 24 and 48 hours – this suggests that values of 2 and 1 would suitable for the P and Q parameters respectively. The SARIMAX model is fit with the parameterisation $(5, 1, 1)(2, 0, 1)$.

4.2.3 Generalised Additive Models

Generalised Additive Models (GAMs) are a flexible extension of linear models that allow for the inclusion of non-linear relationships between predictors and the response variable through smooth functions. In this study, the GAMs are implemented in R using functions in the `mgcv` package (Wood, 2011). The exogenous predictor variables are rescaled to the range $[0, 1]$ to prevent issues arising due to the large difference in scale, particularly between atmospheric pressure and the other variables. For the first

GAM, smooth terms were applied to predictors using thin-plate regression splines, a choice motivated by their computational efficiency and ability to automatically determine the degree of smoothness (Rahman et al., 2018). The models are fitted to the training data using a scaled t-family distribution and identity link function (`scat`, Wood (2011)) – this distribution has heavier tails than the Gaussian distribution (Fernández et al., 1999) which is suitable for this data. The scaled t distribution is parameterised by a scale parameter (σ), determining the spread, and its degrees of freedom (ν), which control the heaviness of the tails – small values of ν correspond to heavier tails and as $\nu \to \infty$, the distribution approaches a Gaussian distribution (Wood, 2011). The parameters are estimated during model fitting using maximum likelihood estimation. Equation 4.4 gives the form of the first model (GAM0):

$$y_t = \beta_0 + s(x_t^{(\mathrm{AP})}) + s(x_t^{(\mathrm{LW})}) + s(x_t^{(\mathrm{NF})}) + s(x_t^{(\mathrm{OW})}) + s(x_t^{(\mathrm{TP})}) + \epsilon_t; \tag{4.4}$$

where β_0 is the intercept term, $s(\cdot)$ is a smooth function (thin-plate regression spline), and $\epsilon_t \sim$ Scaled $\mathrm{t}(\nu, \sigma)$ is the scaled-t residual with ν degrees of freedom and scale, σ.

We extend the GAM to incorporate temporal dependence by adding lagged response values as parametric terms. A ridge (L2) penalty is applied to each of the model coefficients corresponding to the lagged input variables to control the effect magnitude of individual lagged terms. The penalties are implemented using `paraPen` (Wood, 2011). Two different lagged models are implemented. The first uses the lagged response for three hours prior to the forecast – this gives the model information on the current state of the water level. The second lagged model uses the lagged response for 24 hours prior to the forecast – this gives the model information for the last tidal cycle. Equation 4.5 gives the form of the lagged models (GAM3 and GAM24):

$$\begin{aligned} y_t = &\beta_0 + s(x_t^{(\mathrm{AP})}) + s(x_t^{(\mathrm{LW})}) + s(x_t^{(\mathrm{NF})}) + s(x_t^{(\mathrm{OW})}) + s(x_t^{(\mathrm{TP})}) \\ &+ \beta_1 x_t^{(\mathrm{lag}_1)} + \ldots + \beta_K x_t^{(\mathrm{lag}_K)} + \epsilon_t; \end{aligned} \tag{4.5}$$

where β_k is the coefficient for the k^{th} lag term, $k = 1, \ldots, K$ and $K = 3$ for GAM3 and $K = 24$ for GAM24.

4.3 Machine Learning Models

4.3.1 Extreme Gradient Boosting Models

XGBoost, developed by Chen et al. (2016), is a scalable and efficient implementation of gradient boosting. It is applicable to both classification and regression tasks – which can be adapted for time series forecasting by incorporating lagged features, allowing it to capture temporal dependencies in the data. XGBoost models use the concept of ensemble learning, which combines multiple models to create a more accurate and robust model. XGBoost models use (usually shallow) decision trees as the base

model learning units. The trees are trained sequentially, where each additional tree is trained to correct the residual errors of the previous trees. Gradient boosting refers to an iterative process by which the model reduces the loss function by adding new trees fitted to the negative gradient of the loss function with respect to the model's predictions. This process continues until the possible improvement is below some threshold, or until some iteration count limit is reached.

The XGBoost model includes L2 ridge regularisation techniques, which penalise the complexity of the model, as well as subsampling of the observations and features at each iteration to control overfitting. Further hyperparameters such as maximum tree depth and minimum loss reduction threshold are tuned to control the model complexity.

In order to capture trends and temporal dependencies, feature engineering is employed to generate lagged versions of the response variable. Similar to the ARIMA models, autocorrelation and partial autocorrelation functions are used to inform the number of lagged responses required. Our first model includes three lags and the second model 24 lagged response inputs such that $X_t^{(\text{lag}_k)} = Y_{t-k}$, $k = 1, \ldots, K$, where $K = 3$ in the first model, and $K = 24$ in the second. In order to generate these features, the first K observations are dropped from the dataset.

The XGBoost model is implemented using three different feature architectures. The first model (XGB0) has a simple feature structure containing only the five original covariates: atmospheric pressure, Napa flow, local wind, ocean wind, and predicted tide. The second (XGB3) and third (XGB24) models incorporate the temporal aspect of time series prediction by including three and 24 lags (hours) of the response respectively – mirroring the structure of the GAM models above. This allows the models to factor in recent behaviour prior to the forecast. The XGBoost models are implemented in R using the `xgboost` package (Chen et al., 2016). For each model, K-fold cross-validation (K=10) is used to assess the model performance across a range of values for the hyperparameters, while limiting the potential for overfitting to ensure that the model retains the ability to generalise to unseen data. Two rounds of cross-validation grid search are used – the first explores a wide range of values for the hyperparameters, while the second allows for the fine-tuning of the hyperparameters based on the results of the first round.

The learning rate, η, controls the size of the contribution of each sequential tree to the model. The maximum tree depth is the maximum number of levels or nodes possible between root and leaf in each tree – the deeper the tree, the greater the complexity. In each boosting iteration, the algorithm uses a uniform-random subsample of the training data to prevent overfitting – the subsample proportion. Similarly, for each iteration a uniform random subsample of the features is used to build each tree – column sample by tree. This introduces randomness into the model building process to help prevent overfitting and encourage diversity between trees. γ specifies the

minimum loss reduction threshold required to make a partition on a node in a tree; a larger γ value yields smaller trees. The L2 regularisation hyperparameter, λ, controls the strength of the penalty.

TABLE 4.1: Hyperparameter ranges explored and selected values identified through grid search tuning for the XGBoost models.

Hyperparameter	**Range**	**XGB0**	**XGB3**	**XGB24**
Learning Rate (η)	[0.01, 0.1, 0.2, 0.5, 1]	0.1	0.1	0.1
Maximum Tree Depth	[6, 9, 12]	9	9	9
Subsample Proportion	[0.6, 0.8, 0.9, 1]	1.0	0.8	0.8
Column Sample by Tree	[0.6, 0.8, 0.9, 1]	1.0	0.8	0.8
Gamma (γ)	[0, 0.1, 0.5]	0	0	0
Lambda (λ)	[0, 1, 2, 5]	5	5	5

The tuning process uses a grid search to iterate across the hyperparameter space given in Table 4.1, performing 10-fold cross-validation and recording the minimum testing and training RMSE generated under each hyperparameterisation. The final models use the hyperparameters (given in Table 4.1) that result in the lowest testing cross-validation RMSE score. These hyperparameterisations are used to generate out-of-sample forecasts using the testing dataset. Refer to Section 4.4.2 for the detailed methodology of the forecasting process.

4.3.2 Support Vector Machines

The Support Vector Machine (SVM) is a machine learning technique originally developed by Vapnik et al. (1995) for binary classification problems. However, it can be adapted for regression tasks using variants such as epsilon-Support Vector Regression (ϵ-SVR) and nu-Support Vector Regression (ν-SVR). These methods are capable of predicting a continuous target variable using high-dimensional feature spaces and can handle non-linear relationships using kernel functions while minimising both training error and model complexity (Smola et al., 2004). Support Vector Regression (SVR) identifies a subset of the training data as support vectors – only these observations are used to define the regression function, making the model more sparse and efficient. A margin of tolerance is defined – fitted values within this margin are considered satisfactory and are not penalised by the cost function.

In ϵ-SVR, the size of the margin is fixed and explicitly controlled by the ϵ hyperparameter, such that ϵ controls the sensitivity of the model to deviations from the observed value. In ν-SVR, the ν hyperparameter indirectly affects the size of the margin by controlling both the fraction of data points that may become support vectors as well as the upper bound on the fraction of training errors. This grants the model more flexibility by dynamically determining the size of the margin based on the data in compliance with the ν hyperparameter. This method allows one to control the model complexity without requiring explicit prior knowledge about the expected

margin of error, and therefore will be the chosen method for this project. Both the polynomial kernel and the radial basis function (RBF) kernel are suitable options to capture non-linear relationships in the data, but the polynomial kernel can be much more computationally expensive, particularly as the degree increases. Therefore, the RBF kernel is used for this project. The primary optimisation problem for ν-SVR can be formulated as (Chang et al., 2002):

$$\min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\Big(\nu\epsilon + \frac{1}{m}\sum_{i=1}^{m}(\xi_i + \xi_i^*)\Big) \tag{4.6}$$

subject to:

$$\begin{aligned} \mathbf{w}^\top\phi(\mathbf{x}_i) + b - y_i &\leq \epsilon + \xi_i \qquad \forall\ i = 1,\dots,m, \\ y_i - (\mathbf{w}^\top\phi(\mathbf{x}_i) + b) &\leq \epsilon + \xi_i^* \qquad \forall\ i = 1,\dots,m, \\ \xi_i, \xi_i^* &\geq 0 \qquad \forall\ i = 1,\dots,m, \\ \epsilon &\geq 0 \end{aligned} \tag{4.7}$$

where: $\mathbf{w} \in \mathbb{R}^n$ is the weight vector; $b \in \mathbb{R}$ is the bias term; $\mathbf{w}^\top\phi(\mathbf{x}_i) + b$ is the predicted target for input $\mathbf{x}_i$; ϕ is a transformation function; ϵ is the estimated margin of tolerance, determining how much deviation is allowed without penalty; ξ_i and ξ_i^* are slack variables representing deviations from the margin; $C > 0$ is a regularisation hyperparameter controlling the trade-off between model complexity and training error; and $\nu \in [0, 1]$ controls the number of support vectors and training errors allowed.

The optimisation can be reformulated into its dual form, given by (Chang et al., 2002):

$$\min_{\alpha,\alpha_i^*} \quad \frac{1}{2}(\alpha - \alpha^*)^\top\mathbf{Q}(\alpha - \alpha^*) + \mathbf{y}^\top(\alpha - \alpha^*) \tag{4.8}$$

subject to:

$$\begin{aligned} \mathbf{1}^\top(\alpha - \alpha^*) &= 0, \\ \mathbf{1}^\top(\alpha + \alpha^*) &\leq C\nu, \\ 0 \leq \alpha_i, \alpha_i^* &\leq \frac{C}{m} \qquad \forall\, i = 1,\dots,m, \end{aligned} \tag{4.9}$$

where α and α^* are vectors of Lagrange multipliers, $Q_{ij} = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$ is the kernel, and $\mathbf{1}$ is a vector of ones.

After solving the optimisation problem, the predictions for a new input $\mathbf{x}$ can be approximated by (Chang et al., 2002):

$$f(\mathbf{x}) = \sum_{i=1}^{m}(\alpha_i^* - \alpha_i)\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}) + b \tag{4.10}$$

The ν-SVR was implemented in R using the `e1071` library to fit and forecast an SVR model given its hyperparameterisation, and the `mlr` library to facilitate the hyperparameter tuning using cross-validation to minimise RMSE. The cost hyperparameter,

C, controls the trade-off between model complexity and accuracy. As C increases, greater complexity is penalised more strongly. The γ hyperparameter defines the spread of the RBF kernel, controlling the range of influence of a single training point. As γ increases, the area of influence decreases, becoming more responsive to local variation, whereas smaller γ makes the model smoother and less responsive. The ν hyperparameter controls the fraction of margin violations and therefore the number of support vectors generated – a small ν forces a stricter fit to the data, but risks overfitting on the training data.

TABLE 4.2: Hyperparameter ranges explored and selected values identified through grid search tuning for the SVR models.

Hyperparameter	**Range**	**SVR0**	**SVR3**	**SVR24**
Cost (C)	[1, 2.5, 5, 10]	10	10	2.5
Spread (γ)	[0, 0.05, 0.1, 0.5, 1]	0.1	0.1	0.1
Margin Violations (ν)	[0.1, 0.3, 0.5, 0.6]	0.5	0.5	0.6

As with the XGBoost model, the SVR model was implemented using three different feature architectures. The first model has a simple regression structure where the water stage is modelled as a function of the five covariates: atmospheric pressure, Napa flow, local wind, ocean wind, and predicted tide. The second and third models incorporate the temporal aspect of time series prediction by including 3 and 24 lagged features of the response respectively. This allows the model to adapt its prediction based on recent changes in the water level. The hyperparameter tuning results are given in Table 4.2.

4.3.3 Long Short Term Memory Neural Networks

The Long Short-Term Memory (LSTM) network is a deep learning architecture designed for the modelling of sequential data. Adapted from recurrent neural networks, LSTMs allow for the handling of long-term temporal dependencies via memory cells and gating mechanisms (Hochreiter et al., 1997). The memory cell is a storage unit that allows information to be carried forward through the sequence. The gating mechanism controls the flow of information through the memory cell, regulating how much information is 'remembered' and passed on. This combination of functionality allows the LSTM network to maintain long-term dependencies while filtering outdated information.

In order to avoid numerical scale-related issues impacting the model convergence, we rescale the predictor variables to the range $[0, 1]$. We implement two variations of the model – the first model (LSTM3) uses a 3-hour timestep to introduce lags of the response variable into the model and the second model (LSTM24) uses a 24-hour timestep. Each model requires an input feature array to be created with the corresponding timestep. These arrays contain the water level for 3 and 24 hours prior to the time of forecast, and the predictor values at the time of forecast.

The LSTM model architecture consists of an input layer to accept the lagged response values and exogenous predictors, a number of hidden layers, and an output layer to output the 96-hour forecast. The model hyperparameters are tuned using a grid search to identify the best configuration. Table 4.3 outlines the hyperparameter space explored by this process and reports the best combination of values identified for each model. The models are trained using the Adam optimiser and use mean squared error (MSE) as the loss function. The models are fitted using 50 epochs and a 20% validation split.

TABLE 4.3: Hyperparameter ranges explored and selected values identified through grid search tuning for the LSTM networks.

Hyperparameter	**Range**	**LSTM3**	**LSTM24**
Hidden Layers	[1, 2, 3, 4]	2	3
Units	[32, 50, 64, 96, 128]	64	128
Dropout Rate	[0, 0.05, 0.15, 0.25]	0.05	0.05
Batch Size	[32, 64, 128]	32	64
Learning Rate	[0.001, 0.01, 0.1]	0.001	0.001

The model performance is then evaluated on the test dataset. The model is implemented in R using the `keras` (Kalinowski et al., 2024) and `tensorflow` (Allaire et al., 2024) packages for deep learning frameworks.

4.3.4 Variable Importance and Partial Dependence

The statistical models allow for direct interpretation of the model output, with the coefficient values, standard errors, and associated p-values providing information about the variable of interest, including significance, direction, and magnitude of effect on the response. The model fit metrics, such as AIC and R^2, give a relative measure of how well the model fits the data and the proportion of variance in the data captured by the model. By comparison, machine learning models are more opaque to interpret.

The XGBoost framework provides a method to calculate variable or feature importance. This measures the contribution and presence of each feature to the model predictions. It can be used to identify the most influential features and understand the strength of relationship between each feature and the response. This analysis calculates three metrics for each feature – importance, coverage, and frequency. The importance, also called the gain, of a feature measures the contribution of each feature to reducing the loss function across all splits in all the trees. Importance is measured as a proportion of the total reduction, so the sum of the gain across all features equals to one. Feature cover indicates the relative proportion of observations that a feature affects by splitting. Feature frequency is the proportion of occurrences of the feature in the splits across all trees – high frequency indicates that the feature is present for many splits, even if the resulting splits do not contribute significantly

to reducing error. Variable importance is implemented using the `xgb.importance` function in the `xgboost` library (Chen et al., 2016).

Partial dependence plots give a way to visualise the marginal effect of individual features on the model prediction, holding other features constant, and provide insight into the nature of the relationship between the feature and response captured by the model. From this plot we can glean the direction and magnitude of the feature effect, and understand whether the effect on the response is linear or non-linear. The partial dependence plots are generated using the `pdp` library (Greenwell, 2017).

4.4 Out-of-Sample Performance Evaluation

4.4.1 Metric Selection

Root Mean Squared Error (RMSE) is widely regarded as one of the best metrics for evaluating the accuracy of time series prediction models due to its desirable mathematical properties and practical interpretability. RMSE is calculated as follows, $\text{RMSE} = \sqrt{\frac{1}{N}\sum_t^N (y_t - \hat{y}_t)^2}$; where y_t is the observed response at time t, and $\hat{y}_t$ is the corresponding prediction, providing a single value that quantifies the model's prediction error. Mean Absolute Error (MAE) is another common accuracy metric, calculated as $\text{MAE} = \frac{1}{N}\sum_t^N |y_t - \hat{y}_t|$ (Hyndman et al., 2018). Unlike MAE, RMSE penalises larger errors more heavily due to its squaring mechanism, making it particularly sensitive to outliers and large deviations, which are often critical in time series forecasting (Chai et al., 2014). This characteristic is favourable in this application, as large errors could have significant consequences due to unexpected high water levels causing flooding. Willmott et al. (2005) caution that RMSE can be more ambiguous than MAE as RMSE increases with both the variance and total sum of the errors, while MAE only increases with the total sum of the errors. Both RMSE and MAE are scale-dependent and are given in the same units as the response variable, making them intuitive to interpret (Hyndman et al., 2018). Chai et al. (2014) and Willmott et al. (2005) make arguments respectively for and against the use of RMSE for evaluating the accuracy of models – this study will use the RMSE as the primary accuracy metric due to its harsher scoring of large errors; however, the MAE will also be reported for each model.

4.4.2 Approach to Forecasting

The process of generating out-of-sample test predictions is a challenge for time series models. Unlike non-temporal tabular problems, we cannot simply randomly select a proportion of observations from the data set to set aside for testing and validation. As the time series data has intrinsic temporal dependence, we need a contiguous segment of data to perform test forecasts on. We also face the challenge that the models with temporal dependence will handle the prediction for a particular hour differently if it is the first, second, or twenty-fourth hour since the time-of-forecast. In this study,

the last 102 days (2448 hours) of the dataset are set aside for testing on unseen data. The forecast horizon for these short-term forecasts is 96 hours – this matches the forecast horizon used by Largier et al. (2023), which was deemed operationally useful for applications such as evacuation and traffic planning. If we were to divide the test data set into approximately 25 contiguous 96-hour segments, then the observations at times 1, 97, 193,... will always be tested as observations that are one hour since time-of-forecast. Similarly, observations at times 96, 192, ... will always be the final prediction of the forecasts. We will not be able to assess how well the models predict those same observations, with the same covariates, if they occurred at a different time in the forecast period (i.e. a different number of hours since time-of-forecast), with different temporal dependencies. In order to avoid this bias in the test results and to maximise the amount of usage from the data available, the following out-of-sample forecasting strategy is proposed.

Each observation in the test dataset of 2448 observations, from 1 to 2353 (i.e. $2448 - 96$), will be the start of a 96-hour test forecast. This allows us to generate 2353 out-of-sample forecasts on the test dataset (instead of 25), and ensure that every observation (between observation 96 and 2353) is forecast at every time-point from 1 to 96 hours since time-of-forecast. This approach generates a forecast matrix with dimensions 96×2353, where the row denotes hours since time-of-forecast and the column denotes the starting observation in the test dataset. The performance metrics (RMSE, MAE, residual error range) are calculated for each forecast (column) and then aggregated for the entire test set. This forecast matrix also allows us to calculate performance metrics, such as RMSE, for all predictions at particular time-points (number of hours since time-of-forecast). This allows us to assess how the average performance of the forecasts changes as the time horizon increases.

For the temporally-independent models (e.g. linear regression or No lag XGB), each 96-hour forecast is generated in a single action, as the required inputs for all 96 observations are available at the time of forecast. For the models with temporal-dependence (e.g. 3-hour or 24-hour lag XGBoost), the forecast must be generated iteratively, as each predicted value is required as input for the subsequent prediction. The response values from the test dataset cannot be used to generate these lagged response inputs as this would result in data leakage, providing the model with data that it would not have access to in practice. The lagged inputs are initialised using the final observations from the training dataset. These initial inputs are used to predict the output for the first hour of the forecast. That predicted value is then treated as the 1-hour lagged input for the prediction of the following hour, with this process then repeating iteratively for the entire forecast horizon. As the forecast moves further into the future, the amount of uncertainty in the prediction increases as the lagged inputs for the prediction are themselves predicted values.

Chapter 5

Results

This chapter discusses the metrics, primarily root mean squared error (RMSE), used to evaluate the performance of the models in terms of accuracy and variance. We report and tabulate the in-sample training and out-of-sample testing metrics, investigate model accuracy at various time horizons, and visualise forecasted predictions against the corresponding observed values. Where possible, we discuss the interpretation of the model parameters and results to better understand the driving factors of the predictions. We assess which model provides the most accurate predictions, and discuss the effect of incorporating explicit temporal dependency into the models. Summaries of the statistical models' fits can be found in the Appendix.

5.1 Model Metrics and Evaluation

The performance of the models is evaluated using a range of metrics to capture the accuracy – according to the magnitude of the errors – and the consistency, according to the spread of the errors. The models are also evaluated to understand how the forecasts perform at different horizons in the forecast period and to capture potential changes in performance at different points in the tidal cycle. Table 5.1 gives the training and testing performance metrics for all of the models. The Train RMSE, is evaluated on the training dataset and measures the models' ability to capture the information available in the training data, but cannot be used to say anything about the models' ability to capture the underlying patterns and generate predictions on unseen data.

The mean Test RMSE reflects the model's average performance on unobserved data. This Test RMSE score is calculated as the mean value of the RMSE across each of the 96-hour forecasts generated using the testing dataset. Additionally, the table provides the standard deviation of these RMSE values, which gives an indication of the consistency of these error measures throughout the testing data. The mean Test MAE is also reported, as certain attributes of the errors can be inferred from the relationship between RMSE and MAE. RMSE will always be equal to or greater than MAE; if the ratio $\frac{\text{RMSE}}{\text{MAE}}$ is close to 1, it implies a relatively uniform magnitude of

errors, while a ratio greater than 1 indicates the existence of outlying errors in the forecasts.

TABLE 5.1: Model performance metrics (Train RMSE, mean Test RMSE, Standard Deviation of Test RMSE, mean Test MAE, range of testing residual errors) for all models across the entirety of the 96-hour forecast horizon test predictions. All units are in cm.

Model	**Train RMSE**	**RMSE**	**RMSE SD**	**MAE**	**Min Error**	**Max Error**
Naïve Tide	12.4	10.2	2.8	8.0	-36.6	47.8
Linear Regression	10.8	9.6	2.1	7.5	-38.2	44.9
Quad.Regression	10.2	9.8	1.8	7.9	-34.8	40.3
Cubic Regression	9.0	9.1	1.7	7.3	-30.2	36.6
Linear Regr. Seas.	10.5	10.6	2.0	8.7	-35.8	47.2
Quad. Regr. Seas.	10.0	10.5	1.7	8.7	-32.7	42.7
Cubic Regr. Seas	8.8	9.7	1.6	8.0	-27.9	37.9
ARIMAX(5,1,1)	5.8	8.7	1.9	6.5	-42.3	40.2
SARIMAX(5,1,1)(2,0,1)	5.4	8.7	1.9	6.5	-42.3	41.6
GAM No lag	8.8	8.7	1.6	7.1	-28.3	34.5
GAM 3-hour lag	6.5	11.7	2.2	9.3	-31.4	37.5
GAM 24-hour lag	5.2	19.1	2.9	15.9	-49.0	55.3
XGB No lag	2.4	9.0	1.6	7.3	-26.8	31.4
XGB 3-hour lag	0.7	8.5	1.9	6.5	-55.5	34.7
XGB 24-hour lag	0.2	7.5	2.1	5.9	-44.7	42.0
SVR No lag	8.4	8.8	1.6	7.1	-29.8	34.0
SVR 3-hour lag	4.1	9.4	2.5	7.3	-22.7	43.4
SVR 24-hour lag	2.1	9.3	2.8	7.4	-25.9	46.2
LSTM 3-hour lag	21.9	21.7	7.8	17.0	-153.2	152.7
LSTM 24-hour lag	10.7	10.7	3.0	8.2	-76.9	61.5

The best performing model, according to the mean test RMSE, is the Extreme Gradient Boosting (XGBoost) model with 24-hour lag (XGB24) by a margin of 1cm, with a mean test RMSE of 7.5cm. XGBoost with 3-hour lag (XGB3) has the second highest overall accuracy with a mean test RMSE of 8.5cm. This is followed by the ARIMAX and SARIMAX models, as well as Support Vector Regression (SVR0) and Generalised Additive Models (GAM0) with no lagged inputs, which all perform very similarly with mean test RMSEs of 8.7cm and less than a millimeter difference between them. The next most accurate models are the XGBoost model with no lag (XGB0) with a mean RMSE of 9.0cm and cubic regression model (9.1cm).

There are a number of models, including the linear and quadratic regression models with seasonal terms, the Generalised Additive Models (GAMs) with lagged inputs, and the Long Short-Term Memory (LSTM) models, that perform worse than the naïve prediction of estimating water level as the predicted tide. This may be due to sub-optimal model design, such as the seasonal component of the regression models; a lack of adherence to model assumptions, such as having highly correlated lag inputs for the GAM models with 3-hour (GAM3) and 24-hour (GAM24) lagged inputs; or an inability to generate accurate predictions for the entirety of the 96-hour forecast horizon, as is the case for the Long Short-Term Memory (LSTM) models. We will now investigate how the models' performance changes over the forecast period.

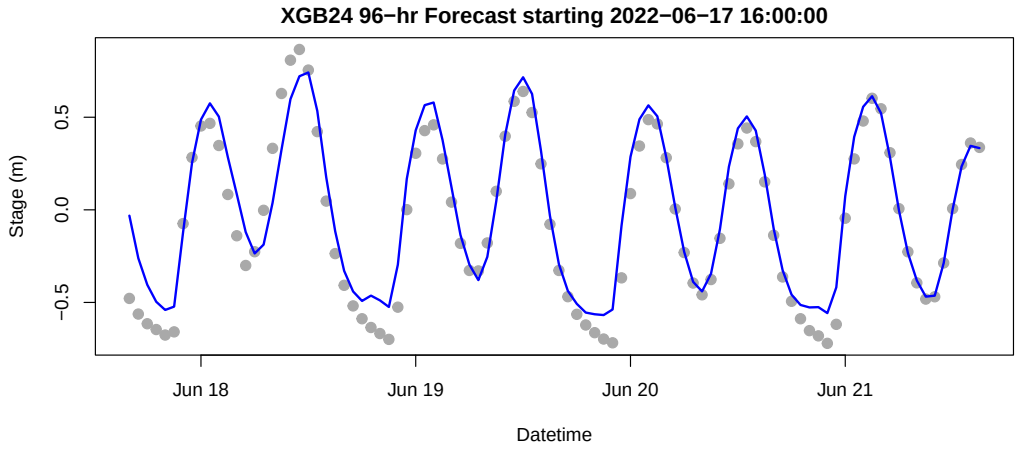

FIGURE 5.1: 96-hour forecast for the XGBoost 24-hour lag model on out-of-sample data. The grey dots denote observed stage level, and the blue line denotes the model predictions.

5.1.1 Accuracy over Time-from-Forecast

Table 5.2 gives the test RMSE of each model, calculated for predictions made at different forecast horizons – 1, 24, 48, 72, and 96 hours. This gives more detail and insight into the performance of the models and a better understanding of how the forecast performance changes over time-from-forecast, as well as assessing which models are better suited to different forecast horizons.

Despite their overall poor average performance seen above, the LSTM models with 24-hour lag (LSTM24) and 3-hour lag (LSTM3) are the most accurate models for a forecast horizon of 1-hour, with test RMSEs of 4.4cm and 5.4cm respectively, followed by the ARIMAX and SARIMAX models (5.9cm) and then XGB24 (8.0cm). However, the accuracy of the LSTM models degrades quickly as the time horizon increases. For 24-hour predictions, LSTM3 is only the 8^{th} most accurate model (9.1cm) and as the forecast horizon increases, it rapidly becomes the most inaccurate model. LSTM24 is more robust to the greater prediction horizons, maintaining its place as the most accurate model for a 24-hour prediction with an RMSE of 6.6cm. However, by the 48-hour prediction, it is the 8^{th} most accurate model (9.2cm), thereafter becoming the 3^{rd} most inaccurate model. These extreme inaccuracies at longer prediction horizons (>48-hours) greatly impact the LSTM models' overall performance metrics in Table 5.1, despite their high level of accuracy in the very short-term (<24-hours). Figure 5.2 shows how the prediction accuracy of LSTM24 degrades over the forecast period.

The ARIMAX and SARIMAX models perform very similarly across all metrics and as such their accuracy will be discussed as if they were a single unit, referred to as the ARIMAX models. They are the second-best performing family of models after the LSTMs for 1-hour predictions (5.9cm), but their accuracy diminishes with increasing forecast horizon, and they are outperformed by the more consistent lagged response XGBoost models – XGB24 at the 24-hour horizon, and XGB3 at the 48-hour

TABLE 5.2: Test accuracy measured by mean RMSE (cm) for all models across different forecasting horizons, measured at 1, 24, 48, 72, and 96 hours.

Model	**1-hour**	**24-hour**	**48-hour**	**72-hour**	**96-hour**
Naïve Tide	10.8	10.7	10.6	10.5	10.5
Linear Regression	10.1	10.0	9.9	9.7	9.7
Quad.Regression	10.1	10.0	9.9	9.9	9.9
Cubic Regression	9.3	9.3	9.2	9.2	9.2
Linear Regr. Seas.	11.0	10.9	10.8	10.7	10.6
Quad. Regr. Seas.	10.7	10.7	10.6	10.5	10.5
Cubic Regr. Seas	9.9	9.8	9.8	9.8	9.8
ARIMAX(5,1,1)	5.9	8.6	8.8	8.9	8.9
SARIMAX(5,1,1)(2,0,1)	5.9	8.6	8.8	8.9	8.9
GAM No lag	8.9	8.9	8.9	8.9	8.9
GAM 3-hour lag	11.9	11.9	11.9	11.9	11.9
GAM 24-hour lag	19.2	19.3	19.3	19.4	19.4
XGB No lag	9.2	9.2	9.2	9.1	9.2
XGB 3-hour lag	8.9	8.7	8.7	8.7	8.7
XGB 24-hour lag	8.0	7.8	7.7	7.7	7.7
SVR No lag	9.0	8.9	8.9	8.9	8.9
SVR 3-hour lag	9.8	9.7	9.7	9.7	9.7
SVR 24-hour lag	9.7	9.7	9.7	9.7	9.8
LSTM 3-hour lag	5.4	9.1	17.5	22.3	27.7
LSTM 24-hour lag	4.4	6.6	9.2	12.6	15.6

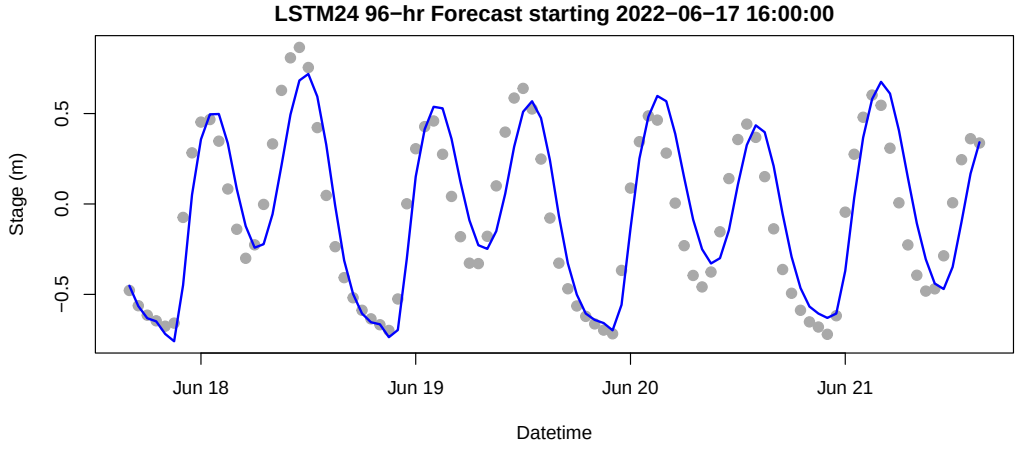

FIGURE 5.2: 96-hour forecast for the LSTM 24-hour lag model on out-of-sample data. The grey dots denote observed stage level, and the blue line denotes the model predictions.

horizon. The ARIMAX models' accuracy is much more robust to time-from-forecast than those of the LSTM models, resulting in the ARIMAX models outperforming the LSTM models as the forecast horizon increases.

The XGBoost models with lagged response inputs are the best performing models overall, and the prediction accuracy is remarkably consistent, even at the greater

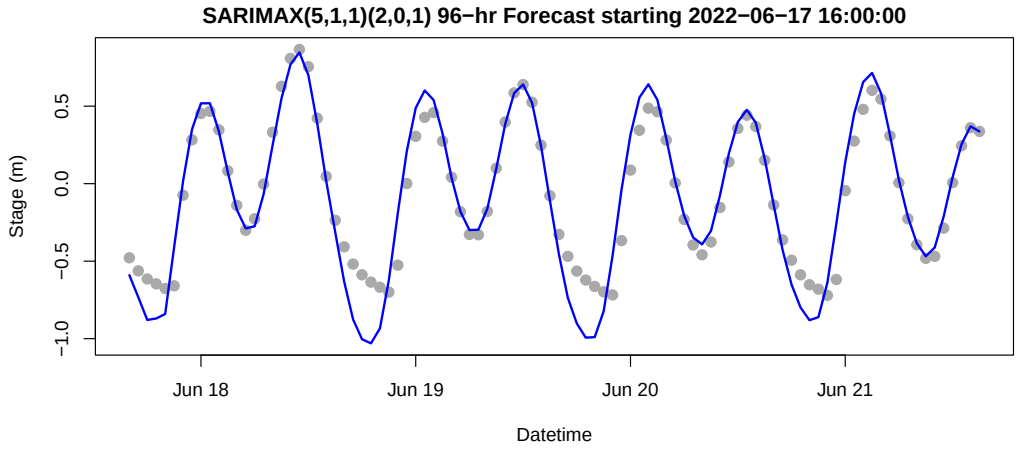

FIGURE 5.3: 96-hour forecast for the SARIMAX(5,1,1)(2,0,1) model on out-of-sample data. The grey dots denote observed stage level, and the blue line denotes the model predictions.

forecast horizons. This is notable as these XGBoost models are generating predictions using previous predictions as input – which one would expect to introduce greater variance to the output. XGB24 is the highest accuracy model for prediction horizons of 48 hours and greater. SVR0 and GAM0 are also very consistent and maintain their average performance over forecast horizons with RMSEs of 8.9cm – this not surprising as these models do not have any temporal dependence. Likewise, the regression models are fairly consistent over the increasing forecast horizons – although we do see the same slight increase in accuracy at longer forecast horizons as discussed for Table 3.4.

The cubic polynomial is the best performing of the regression models with a mean RMSE of 9.1cm, see Table 5.1, a similar accuracy to XGB0. The linear and quadratic models are less accurate by a margin of 0.5cm and 0.7 cm respectively. The seasonal regression models are less accurate on the test data across the board – the addition of the seasonal terms is discussed further in Section 5.1.3.

As shown in Section 3.2, although the error correction (EC) improves the accuracy of the replicated regression (RWL-regression) models when predicting the residual water level (RWL), this improvement in accuracy is lost when the TP is added back to generate predictions of the stage level – in addition the worst-case errors increase by 6cm for the EC models. The accuracy of the RWL-regression models for their stage level predictions is similar across all polynomial-orders and error correction variants. Comparing this performance to the subsequent stage-regression models that predict the stage level directly, we see that the linear and quadratic stage-regression models are less accurate than their RWL-regression counterparts when predicting stage level. The stage-regression model with the greatest accuracy, the cubic polynomial with no seasonal component, achieves a similar overall average test RMSE relative to any of the RWL-regression models, including those with error correct (EC).

TABLE 5.3: Test accuracy measured by mean RMSE (cm) for all models at Positive Stage and Negative Stage.

Model	**Positive Stage**	**Negative Stage**
Naïve Tide	8.3	12.2
Linear Regression	8.0	10.5
Linear Regr. Seas.	9.5	11.1
Quad.Regression	9.3	9.7
Quad. Regr. Seas.	10.2	10.2
Cubic Regression	9.0	10.0
Cubic Regr. Seas	9.5	10.1
ARIMAX(5,1,1)	6.1	10.5
SARIMAX(5,1,1)(2,0,1)	6.1	10.5
GAM No lag	7.8	9.3
GAM 3-hour lag	11.6	11.4
GAM 24-hour lag	21.8	15.5
XGB No lag	8.0	9.7
XGB 3-hour lag	8.4	8.3
XGB 24-hour lag	7.1	7.6
SVR No lag	8.8	9.2
SVR 3-hour lag	7.8	9.5
SVR 24-hour lag	9.4	9.3
LSTM 3-hour lag	18.3	24.0
LSTM 24-hour lag	11.9	9.0

5.1.2 Positive and Negative Stage

As the context of this project revolves around the risk of flooding, it is important to assess the accuracy of the models when the water level is high, as it is unlikely that flooding will occur at low tide. To evaluate this performance, Table 5.3 gives the mean RMSE of the test predictions where the observed stage is greater than zero (positive stage), and the mean RMSE where the stage is equal to or below zero (negative stage). Note that the stage level is measured relative to some arbitrary reference point, allowing it be both positive (above) and negative (below). The ARIMAX models have the greatest accuracy (6.1cm) for the positive stage, followed by the XGB24 (7.1cm). The majority of the models are more accurate for the positive stage observations, LSTM24 and GAM24 are the only two models whose predictions are substantially more accurate for negative stage observations relative to the positive stage. In Figure 5.2, we can see high level of accuracy for LSTM24 on the negative stage predictions, particularly in the first 48 hours of the forecast period.

LSTM3 and the ARIMAX models have the greatest increases in accuracy between the negative stage and positive stage predictions. Figure 5.3 shows how the ARIMAX models struggle to capture the low-tide dynamics. The naïve tidal prediction also has a large difference between the positive and negative stages – as mentioned previously, the uTide routine for tidal prediction performs poorly at low tide as it cannot capture the drainage dynamics present at Novato Creek mouth.

It is interesting to note the difference in performance of the polynomial regression models – the linear regression models have greater differences in performance between positive and negative stages as the linear terms are unable to capture the low tide dynamics. By comparison, the quadratic and cubic regression models experience little-to-no difference in performance between the stage levels as their increased complexity affords them more flexibility to capture the non-linear relationship between tidal prediction (TP) and the response stage level at low-tide. Figure 5.4 showcases the difference in forecast accuracy, particularly in the troughs of the time series, between the linear and cubic multiple polynomial regression models.

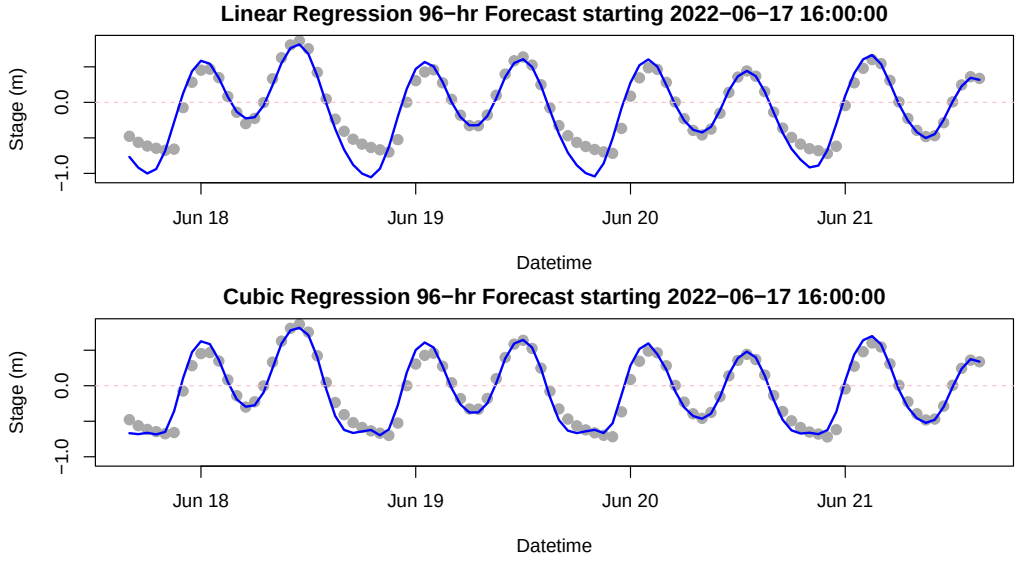

FIGURE 5.4: Comparison of 96-hour forecast accuracy on out-of-sample data for the Linear and Cubic Multiple Polynomial Regression models. The grey dots denote observed stage level, the blue lines denotes the model predictions, and the pink dashed line at zero separates the positive and negative stage observations.

5.1.3 Variable Importance and Interpretation

Among other things, we want to understand how the addition of lagged reponses features as inputs impacts the model performance. The XGBoost forecast accuracy improves notably with the addition of lagged response inputs. Including three features of the lagged response leads to a decrease in mean test RMSE relative to XGB0 of 0.5cm, while adding lag inputs for 24 hours decreases the mean test RMSE by a further 1.0cm for a total 1.5cm improvement, see Table 5.1. In addition, this improvement in accuracy is consistent over the entire 96-hour forecast horizon, see Table 5.2. The LSTM network also generates more accurate forecasts when provided with a greater number (24 vs 3 hours) of lag inputs. LSTM24, relative to LSTM3, decreases the mean 1-hour prediction RMSE by 1.0cm, decreases the 24-hour RMSE by 2.5cm, and allows the model to be significantly more robust to longer prediction horizons as LSTM24 improves on the mean RMSE of LSTM3 for prediction horizons of 48, 72, and 96 hours by 8.3cm, 9.7cm, and 12.1cm respectively, see Table 5.2.

By contrast, the SVR model test accuracy suffers when including additional inputs of the lagged response. The 3-hour and 24-hour lag SVR models (SVR3 and SVR24) perform similarly, increasing the mean test RMSE by 0.6cm and 0.5cm respectively relative to SVR0. This is likely due to overfitting, as the training RMSE of the SVR models decreases substantially with the addition of the lag inputs. The lagged values may introduce dependencies that are specific to the training data but do not generalise well. There is a similar dynamic for GAM models, where adding the additional lag terms to the model improves the training accuracy, but severely reduces the test accuracy. Despite adding L2 regularisation to penalise the lag term coefficients in the GAM models, the magnitude of these coefficients is only fracionally smaller relative to a non-regularised model. This, combined with the p-values of the lag terms ($< 2 \times 10^{-16}$), suggests that the model views the lag terms as highly significant and strongly correlated with the response. In this case, the penalty will not substantially shrink the coefficients as the likelihood improvement outweighs the penalty during the maximum likelihood estimation.

TABLE 5.4: Regression coefficient p-values for MLR seasonal components

	Linear Seas.	**Quad. Seas.**	**Cubic Seas.**
Lunar Sine	9.43×10^{-13}	6.04×10^{-9}	5.72×10^{-2}
Lunar Cosine	$< 2 \times 10^{-16}$	$< 2 \times 10^{-16}$	$< 2 \times 10^{-16}$
Annual Sine	6.77×10^{-3}	2.02×10^{-1}	4.17×10^{-4}
Annual Cosine	$< 2 \times 10^{-16}$	$< 2 \times 10^{-16}$	$< 2 \times 10^{-16}$

The trigonometric seasonality components of the stage regression models are seen to be significant (with the exception of the annual sine curve in the quadratic model) as per the p-values in Table 5.4, and the addition of them to the models decreases the AIC and improve the training RMSE by 0.2cm for all three models, see Table 5.1. However, this improvement does not carry over to the unseen testing data. The mean test RMSE of the regression models increases by 1.0cm for the linear model, by 0.7cm for the quadratic model, and by 0.6cm for the cubic model relative to the corresponding non-seasonal models. This indicates that the addition of these seasonal components leads to overfitting as the models use the additional flexibility of the additional parameters to fit to noise in the training data. The Generalised Additive Models (GAMs) perform very poorly with the addition of lagged inputs. This is likely due to either the strong dependence between the adjacent lagged inputs or overfitting to the training data due to the strong correlation between the lagged inputs and the response. Further work is required to better understand this dynamic and properly implement GAMs with temporal components.

For the multiple regression models, the relationship between the response and the predictor variables can be interpreted from the regression coefficients. Particularly for the linear model, this interpretation is simple and intuitive – the sign of the

coefficient shows the direction of the relationship, which is negative or inverse for atmospheric pressure (AP) and a positive association for the other predictors. The variable importance can also be evaluated by assessing the change in R^2 due to the addition of that variable. The regression models are driven primarily by the tidal prediction (TP) covariate. Of the meteorological predictors in the linear model, AP gives the greatest increase in R^2 (+0.0054), followed by Napa flow (NF, +0.0027) and ocean wind (OW, +0.0025), with local wind (LW) providing the smallest improvement (+0.0002). The addition of the seasonal terms to the linear model increases R^2 by +0.0027 – see Table B.2.

Table 5.5 reports the importance, coverage, and frequency of the XGBoost models' features to provide insight into the value that each feature contributes. Feature importance measures the proportional contribution of each feature to the total reduction of the loss function – the greater the importance, the more critical the feature is to the model's performance. As expected, all three XGBoost models are dominated by the tidal prediction feature. In XGB0, the importance of the non-tidal exogenous features, in descending order, are AP, NF, OW, and LW. This is consistent with the R^2 analysis of the linear regression model.

In XGB3, lag features have greater importance than the exogenous features, with the one- and three-hour lags being the more influential. For XGB24, the 24-hour and 1-hour lag features have the greatest importance following the tidal prediction – all other features for this model have very low importance (> 0.005). Feature coverage indicates the relative proportion of observations affected by the feature. In XGB0, TP has the most coverage, but both OW and LW also have substantial coverage, accounting for almost half of the observations. For XGB3, the lag features have the most coverage, with the one- and two-hour lags having greater coverage than the tidal prediction. Similarly for XGB24, the one-hour lag has the greater coverage, followed by tidal prediction, the 24-hour, 3-hour, and 2-hour lags, local wind and ocean wind. The frequency of features is relatively evenly split for all three models, with NF having the lowest frequency across the board, and the 1-hour lag has the greatest frequency for both XGB3 and XGB24.

Partial dependence plots visualise the marginal effect of individual features on the XGBoost model predictions, while holding other features constant, and provide insight into the nature of the relationship between the feature and response captured by the model. The first plot in Figure 5.5 shows that AP has a negative relationship with the stage level, while the second plot shows a positive relationship between stage and TP. Both of these effects are as expected due to the respective barometer and tidal effects on the water level. The effect of TP is of a much greater magnitude than that of AP, as indicated by the range of partial dependence. The AP and TP curves for XGB0 are substantially steeper than those of the Lag models, indicating a stronger effect on the response – in lieu of the lagged inputs, the model relies more heavily on the exogenous predictors. XGB0, in particular, has recreated the dynamic between

TABLE 5.5: XGBoost Feature Importance, Cover, and Frequency for XGB0, XGB3, and XGB24.

Feature	Importance			Cover			Frequency		
	XGB0	XGB3	XGB24	XGB0	XGB3	XGB24	XGB0	XGB3	XGB24
TP	0.936	0.752	0.853	0.302	0.149	0.083	0.196	0.118	0.043
AP	0.020	0.005	0.004	0.100	0.055	0.027	0.236	0.119	0.045
LW	0.012	0.002	0.001	0.242	0.101	0.036	0.228	0.122	0.048
NF	0.017	0.004	0.002	0.128	0.070	0.027	0.143	0.082	0.034
OW	0.015	0.004	0.003	0.228	0.101	0.034	0.198	0.114	0.049
Lag 1		0.193	0.053		0.195	0.130		0.178	0.097
Lag 2		0.009	0.004		0.186	0.039		0.142	0.051
Lag 3		0.032	0.001		0.143	0.045		0.125	0.041
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Lag 23			0.002			0.032			0.028
Lag 24			0.070			0.065			0.039

TP and stage seen in Figure 3.4, highlighting XGBoost's ability to capture non-linear relationships. Naturally, the third plot in Figure 5.5 shows a positive relationship between predicted stage and the 1-hour lag feature. The magnitude of the 1-hour lag feature effect is similar to that of TP, however, the effect of the 1-hour lag input is stronger in XGB3 than XGB24, where the effects of the lagged inputs are more diluted and spread between the 24 features.

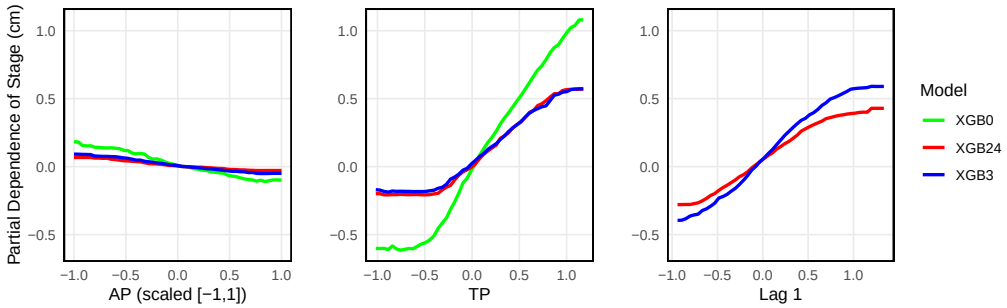

FIGURE 5.5: Partial Dependence Plots for the AP, TP, and Lag 1 inputs of the XGBoost models. XGB0, XGB3, and XGB24 are plotted in green, blue, and red respectively.

This chapter gives the results of the test data forecasts for each of the six families of models, with the primary metric for accuracy being RMSE. The XGB24 performs the best on average across the entire 96-hour forecast horizon. The ARIMAX and LSTM models are the most accurate models for a 1-hour forecast, with the 24-hour lag LSTM giving the most accurate predictions for the first 24 hours of the forecast. Many of the models display a substantial difference in accuracy for observations in the positive and negative stage. The ARIMAX models are, on average across the forecast

horizon, the most accurate for the positive stage. The regression models' coefficients alongside the variable importance and partial dependence of the XGBoost models show that, as expected, TP is the primary driving factor of the forecast with the other exogenous variables providing small adjustments to the predicted value. These findings highlight the differences between the families of models as well as between different architectures or parameterisations of models in the same family, in terms of accuracy, consistency, and complexity. The results provide a basis for the Discussion in Chapter 6.

Chapter 6

Discussion

6.1 Summary

The primary aim of this study is to develop a computationally efficient, site-specific model capable of generating accurate, short-term forecasts of the coastal water level. We implement and evaluate the performance of three families of statistical models – multiple regression, Autoregressive Integrated Moving Average with Exogenous variables (ARIMAX), and Generalised Additive Models (GAMs) – and three machine learning methods – Extreme Gradient Boosting (XGBoost), Support Vector Regression (SVR), and Long Short-Term Memory (LSTM) networks. The site of interest for the development of the models is Novato Creek mouth on the north shore of San Francisco Bay, California.

The existing work at this site aimed to model the residual water level (RWL), after removing the tidal effect (measured as the uTide tidal prediction (TP)) from the stage level, using a set of four exogenous variables – atmospheric pressure (AP), the south-east local onshore wind (LW), the Napa river flow (NF), and the south-southwest wind over the ocean (OW). This project adjusts the model structure to model the stage water level as a function of those four exogenous variables and includes TP as an additional input variable. This change gives the models more flexibility and allows the more complex models to capture non-linear relationships between TP and the stage level, particularly at low tide where TP is less robust. It also avoids the addition of unnecessary uncertainty into the data through the calculation of the residual water level variable.

The regression models – which entail linear regression and polynomial regression of quadratic and cubic orders – also include three 'seasonal' variants. These models have additional trigonometric sine and cosine terms with periods corresponding to the moon's orbit of the Earth and the Earth's orbit of the Sun. The non-seasonal cubic regression model provides the most accurate predictions, measured by RMSE and MAE, as the increased complexity of higher-order polynomials allows the model to capture non-linear relationships. The 'seasonal' terms improve the model fit on

training data, but decrease forecast accuracy on unseen data – likely due to overfitting to the training data. The forecast accuracy is stable over the 96-hour forecast horizon, but the model performs significantly better when forecasting positive stage observations.

The XGBoost model with 24-hour lagged inputs (XGB24) is, on average, the most accurate model across the entire 96-hour forecast horizon and throughout the tidal cycle. The LSTM network with 24-hour lagged inputs generates the most accurate predictions in the very short-term, up to 24 hours from time of forecast. The different parameterisations of the ARIMAX model have almost identical test accuracy and are, on average, the most accurate for positive stage observations across the forecast horizon – but even these predictions are less accurate than the 24-hour LSTM positive stage predictions for the first 24 hours from time of forecast.

6.2 Reflections

We inherited, via the dataset, the difficulties of robustly calculating RWL, given that the uTide tidal prediction struggles to capture the drainage dynamic at low water levels. This leads to a significant increase in variance and error when moving from RWL-scale to stage-scale predictions. Changing the structure to model the stage level directly, using TP as one of the predictor variables gives each model more agency to flexibly model the non-linearities between the two. However, this approach reduces the ability to interpret the model output to understand the effects of the non-tidal predictors.

Using personal hardware to train and tune the machine learning models limited our ability to thoroughly interrogate possible model architectures and explore the hyperparameter space. For example, it is possible that the LSTM models could be improved by exploring more complex model architectures with increased hidden layers and units. The SVR model in particular was very computationally expensive – the tuning process required more than 135 hours of run time. This made it difficult to explore the hyperparameter space with sufficient range and a sufficiently fine resolution, with the cost hyperparameter being exponentially expensive in terms of computation time, making exploring large values infeasible.

6.3 Future Research

The XGBoost and LSTM models exemplify how including lagged response values as inputs can have a large positive effect on the forecast accuracy as the models can incorporate the relationship between the recent historical water levels and the subsequent levels. Similarly, there is possibility for further improvement to forecast accuracy by incorporating lagged values of the exogenous variables as additional features. This would allow the models to capture potential relationships or interactions between the

exogenous variables and the response as the exogenous variables change. The choice of lagged inputs to include could be informed by knowledge of the expected physical effects. For example, the wind effect (the piling up of water against the shore due to wind) may be stronger if the South-East wind has been blowing strongly for multiple hours continuously than if a strong wind has just started.

The LSTM models show a lot of promise with their extremely high test accuracy for very short-term predictions ($\leq$ 24-hour horizon), however the degradation of accuracy as the forecast horizon increases limits the model as a good option for this application. It is possible that accurate horizon for the LSTM models could be extended with greater computing power and more complex models with more parameters. The field of machine learning is also constantly evolving, and there are, and likely will be many more, cutting-edge machine learning methods that could be applied to this problem.

Some of the models differed substantially, in terms of accuracy, at the peaks (high-tide, positive stage) and troughs (low-tide, negative stage). For example, the ARIMAX models are much more accurate for the positive stage predictions than the negative stage, while the LSTM24 model captures the troughs more precisely than it does the peaks. As the context of the forecast relates to flooding, we are more concerned with accurately forecasting the peaks than the troughs. In pursuit of this aim, one could explore adjusting the cost functions or training objectives to incentivise the models to prioritise accuracy at the peaks, even if it comes at the detriment of the trough-accuracy. The technical implementation of this would differ from model to model.

There are many inputs from the literature that would be useful for future extensions to this work that are beyond the scope of this study. A Bayesian approach would provide the full distribution of parameters alongside the point-forecast, quantifying the uncertainty in the model (Harrison et al., 1971). This could be used to generate the probability of the forecast water level exceeding some threshold (Brooks et al., 2008), which would be very useful when moving from a water level forecast to quantifying the risk of flood events. A number of the papers in literature suggest using hybrid or ensemble models that combine different models or architectures to improve model accuracy – these include Tsakiri et al. (2018)'s hybrid MLR-ANN model, Han et al. (2008)'s neural network ensemble, Cho et al. (2022)'s LSTM-GRU hybrid models, and Wang et al. (2020)'s discrete wavelet transformation autoregressive integrated moving average grid search optimised XGBoost (DWT-ARIMA-GSXGB) hybrid ensemble model.

6.4 Conclusion

XGB24 is the best all-rounder choice as it has the highest overall test accuracy, it is the second most accurate model for the positive stage predictions and the most

accurate for negative stage predictions, and it is able to make predictions with consistent accuracy over the entire 96-hour forecast horizon. The XGBoost models also provide opportunity for interpretation through the variable importance and partial dependence plots, which measure the contribution of each feature and visualise the effect of it on the response.

The ARIMAX models are also a strong option with the second highest overall test accuracy, and the best accuracy for positive stage predictions. The performance degrades slightly for the longer forecast horizons, but this is not detrimental to the model performance. These models also provide good interpretability through the model parameter estimates which give the effect size and associated variance for each of the autoregressive, moving average, seasonal, and exogenous terms.

LSTM networks, as implemented in this study, could be a good option for more niche applications that require forecasts of one day or less, as they generate the most accurate predictions for the first 24-hours from time of forecast. However, the deterioration in accuracy for longer horizons deters us from recommending them for the defined aim of generating accurate multi-day (96-hour) forecasts.

Meteorological covariates provide information that contributes significantly to the models' ability to forecast the water level. This is evident from the statistically significant coefficients in the statistical methods (regression, ARIMAX, GAM), and the non-zero contributions shown in the XGBoost feature importance and partial dependence. These model results also provide interpretation of the feature effects – giving an estimate of the magnitude and direction, and for the statistical models, a measure of the variance of the effect size.

Appendix A

Methods

The R scripts to implement all six families of models, as well as the replication models, are available at: https://github.com/jplharrison/msc_coastal_flooding

A.1 Statistical Models

A.1.1 Polynomial Regression

$$y = \beta_0 + \beta_1 x_{\text{AP}} + \beta_2 x_{\text{LW}} + \beta_3 x_{\text{NF}} + \beta_4 x_{\text{OW}} + \beta_5 x_{\text{TP}} + \epsilon \tag{A.1}$$

$$\begin{aligned} y = {} & \beta_0 + \beta_1 x_{\text{AP}} + \beta_2 x_{\text{LW}} + \beta_3 x_{\text{NF}} + \beta_4 x_{\text{OW}} + \beta_5 x_{\text{TP}} \\ & + \beta_6 x_{\text{AP}}^2 + \beta_7 x_{\text{LW}}^2 + \beta_8 x_{\text{NF}}^2 + \beta_9 x_{\text{OW}}^2 + \beta_1 0 x_{\text{TP}}^2 \\ & + \beta_{11} x_{\text{AP}} x_{\text{LW}} + \beta_{12} x_{\text{AP}} x_{\text{NF}} + \beta_{13} x_{\text{AP}} x_{\text{OW}} + \beta_{14} x_{\text{AP}} x_{\text{TP}} \\ & + \beta_{15} x_{\text{LW}} x_{\text{NF}} + \beta_{16} x_{\text{LW}} x_{\text{OW}} + \beta_{17} x_{\text{LW}} x_{\text{TP}} + \beta_{18} x_{\text{NF}} x_{\text{OW}} \\ & + \beta_{19} x_{\text{NF}} x_{\text{TP}} + \beta_{20} x_{\text{OW}} x_{\text{TP}} + \epsilon \end{aligned} \tag{A.2}$$

$$\begin{aligned} y = {} & \beta_0 + \beta_1 x_{\text{AP}} + \beta_2 x_{\text{LW}} + \beta_3 x_{\text{NF}} + \beta_4 x_{\text{OW}} + \beta_5 x_{\text{TP}} \\ & + \beta_6 x_{\text{AP}}^2 + \beta_7 x_{\text{LW}}^2 + \beta_8 x_{\text{NF}}^2 + \beta_9 x_{\text{OW}}^2 + \beta_{10} x_{\text{TP}}^2 \\ & + \beta_{11} x_{\text{AP}}^3 + \beta_{12} x_{\text{LW}}^3 + \beta_{13} x_{\text{NF}}^3 + \beta_{14} x_{\text{OW}}^3 + \beta_{15} x_{\text{TP}}^3 \\ & + \beta_{16} x_{\text{AP}} x_{\text{LW}} + \beta_{17} x_{\text{AP}} x_{\text{NF}} + \beta_{18} x_{\text{AP}} x_{\text{OW}} + \beta_{19} x_{\text{AP}} x_{\text{TP}} \\ & + \beta_{20} x_{\text{LW}} x_{\text{NF}} + \beta_{21} x_{\text{LW}} x_{\text{OW}} + \beta_{22} x_{\text{LW}} x_{\text{TP}} + \beta_{23} x_{\text{NF}} x_{\text{OW}} \\ & + \beta_{24} x_{\text{NF}} x_{\text{TP}} + \beta_{25} x_{\text{OW}} x_{\text{TP}} + \beta_{26} x_{\text{AP}}^2 x_{\text{LW}} + \beta_{27} x_{\text{AP}}^2 x_{\text{OW}} \\ & + \beta_{28} x_{\text{LW}}^2 x_{\text{AP}} + \beta_{29} x_{\text{LW}}^2 x_{\text{OW}} + \beta_{30} x_{\text{NF}}^2 x_{\text{AP}} + \beta_{31} x_{\text{NF}}^2 x_{\text{LW}} \\ & + \beta_{32} x_{\text{NF}}^2 x_{\text{OW}} + \beta_{33} x_{\text{OW}}^2 x_{\text{LW}} + \beta_{34} x_{\text{OW}}^2 x_{\text{TP}} + \beta_{35} x_{\text{TP}}^2 x_{\text{AP}} \\ & + \beta_{36} x_{\text{TP}}^2 x_{\text{LW}} + \beta_{37} x_{\text{TP}}^2 x_{\text{OW}} + \beta_{38} x_{\text{AP}} x_{\text{LW}} x_{\text{NF}} \\ & + \beta_{39} x_{\text{AP}} x_{\text{LW}} x_{\text{OW}} + \beta_{40} x_{\text{AP}} x_{\text{LW}} x_{\text{TP}} + \beta_{41} x_{\text{AP}} x_{\text{NF}} x_{\text{TP}} \\ & + \beta_{42} x_{\text{AP}} x_{\text{OW}} x_{\text{TP}} + \beta_{43} x_{\text{LW}} x_{\text{NF}} x_{\text{OW}} + \beta_{44} x_{\text{LW}} x_{\text{NF}} x_{\text{TP}} \\ & + \beta_{45} x_{\text{NF}} x_{\text{OW}} x_{\text{TP}} + \epsilon \end{aligned} \tag{A.3}$$

Appendix B

Results

B.1 Statistical Models

B.1.1 ARIMA

Table B.1: ARIMA and SARIMA model coefficients (and standard errors), rounded to three decimal places.

	ARIMA(5,1,1)	SARIMA(5,1,1)(2,0,1)
AR1	1.063 (0.006)	0.811 (0.007)
AR2	-0.602 (0.008)	-0.274 (0.009)
AR3	0.231 (0.009)	0.080 (0.008)
AR4	-0.115 (0.008)	-0.056 (0.007)
AR5	-0.041 (0.006)	-0.048 (0.006)
MA1	-0.963 (0.002)	-0.971 (0.002)
SAR1		0.419 (0.027)
SAR2		-0.201 (0.012)
SMA1		0.074 (0.027)
AP	-0.363 (0.012)	-0.373 (0.011)
LW	0.016 (0.005)	0.024 (0.004)
NF	0.334 (0.030)	0.252 (0.024)
OW	0.082 (0.007)	0.078 (0.006)
TP	1.018 (0.002)	1.020 (0.002)

B.1.2 Polynomial Regression

TABLE B.2: Polynomial Regression Model Coefficients (and corresponding p-values) for stage-regression models, rounded to three decimal places – 0.000 implies < 0.0005. Adjusted R^2 measures explanatory power of the model.

	Linear	Quad.	Cubic	Linear Seas.	Quad. Seas.	Cubic. Seas.
Intercept	0.166(0.000)	0.195(0.000)	0.327(0.000)	0.195(0.000)	0.171(0.000)	0.233(0.000)
Lunar Sin				−0.564(0.000)	0.005(0.000)	0.001(0.057)
Lunar Cos				−0.077(0.000)	0.009(0.000)	0.008(0.000)
Annual Sin				0.139(0.000)		−0.003(0.000)
Annual Cos				1.091(0.000)	0.031(0.000)	0.030(0.000)
AP	−0.305(0.000)	−0.564(0.000)	−1.146(0.000)	0.256(0.000)	−0.379(0.000)	−0.447(0.000)
LW	0.037(0.000)	−0.077(0.000)	0.308(0.000)	−0.067(0.000)	−0.046(0.002)	0.333(0.000)
NF	0.602(0.000)	1.436(0.000)	1.811(0.000)	−1.292(0.000)	1.273(0.000)	1.550(0.000)
OW	0.102(0.000)	0.139(0.000)	−0.064(0.058)	0.065(0.000)	0.131(0.000)	
TP	1.015(0.000)	1.091(0.000)	1.297(0.000)	0.163(0.000)	1.096(0.000)	1.298(0.000)
AP^2		0.256(0.000)	1.092(0.000)			−0.312(0.089)
LW^2		−0.067(0.000)	0.041(0.438)		−0.040(0.000)	
NF^2		−1.292(0.000)	−4.343(0.000)		−1.157(0.000)	−3.331(0.000)
OW^2		0.065(0.000)	0.054(0.000)		0.060(0.000)	
TP^2		0.125(0.000)	−0.047(0.000)		0.122(0.000)	−0.042(0.000)
AP^3			−0.445(0.001)			0.348(0.003)
LW^3			−0.141(0.000)			−0.116(0.000)
NF^3			3.866(0.000)			3.372(0.000)
OW^3			0.099(0.000)			0.049(0.019)
TP^3			−0.406(0.000)			−0.406(0.000)
AP · LW		0.163(0.000)	−1.16(0.000)		0.075(0.006)	−1.302(0.000)
AP · NF		0.340(0.001)	1.055(0.000)		0.316(0.002)	
AP · OW		−0.042(0.081)	0.621(0.000)		−0.056(0.013)	0.360(0.000)
AP · TP		−0.131(0.000)	−0.194(0.000)		−0.141(0.000)	−0.199(0.000)
LW · NF		0.987(0.000)	0.452(0.067)		0.935(0.000)	
LW · OW		0.035(0.009)	−0.044(0.398)		−0.047(0.001)	−0.056(0.123)
LW · TP		0.016(0.020)	0.207(0.000)		0.014(0.031)	0.187(0.000)
NF · OW		−0.860(0.000)	−0.637(0.000)		−0.722(0.000)	−0.323(0.002)
NF · TP		−0.193(0.000)	−0.486(0.000)		−0.197(0.000)	−0.500(0.000)
OW · TP		0.072(0.000)	0.01(0.674)		0.066(0.000)	
AP^2 · LW			1.12(0.000)			1.223(0.000)
AP^2 · NF						1.817(0.000)
AP^2 · OW			−0.532(0.000)			−0.296(0.000)
LW^2 · AP			−0.302(0.002)			−0.161(0.000)
LW^2 · OW			0.35(0.000)			0.326(0.000)
NF^2 · AP			−3.014(0.000)			−3.366(0.000)
NF^2 · LW			−0.692(0.044)			
NF^2 · OW			0.694(0.014)			
OW^2 · AP						0.100(0.000)
OW^2 · LW			−0.164(0.000)			−0.074(0.090)
OW^2 · NF						0.276(0.080)
OW^2 · TP			−0.052(0.000)			−0.052(0.000)
TP^2 · AP			0.398(0.000)			0.383(0.000)
TP^2 · LW			−0.06(0.000)			−0.054(0.000)
TP^2 · OW			−0.112(0.000)			−0.120(0.000)
AP · LW · NF			0.793(0.026)			1.184(0.000)
AP · LW · OW			0.303(0.002)			0.213(0.006)
AP · LW · TP			−0.345(0.000)			−0.311(0.000)
AP · NF · TP			−0.612(0.006)			−0.487(0.023)
AP · OW · TP			0.089(0.055)			0.098(0.000)
LW · NF · OW			−0.593(0.002)			−0.775(0.000)
LW · NF · TP			−1.309(0.000)			−1.165(0.000)
NF · OW · TP			1.086(0.000)			1.024(0.000)
Adjusted R^2	0.946	0.952	0.963	0.949	0.954	0.964

B.1.3 GAM

TABLE B.3: Summary of GAM Model Results

Metric	**Model 1**	**Model 2**	**Model 3**
Scaled t parameter estimates			
(ν, σ)	(5.854, 0.072)	(3, 0.04)	(3, 0.03)
Parametric Coefficients: Estimate (p-value)			
Intercept	0.0046 ($< 2e^{-16}$)	-0.0005 (0.0608)	-0.0033 ($< 2e^{-16}$)
Lag 1	—	1.441 ($< 2e^{-16}$)	1.150 ($< 2e^{-16}$)
Lag 2	—	-1.163 ($< 2e^{-16}$)	-0.675 ($< 2e^{-16}$)
Lag 3	—	0.321 ($< 2e^{-16}$)	0.172 ($< 2e^{-16}$)
Lag 4	—	—	-0.056 ($< 2e^{-16}$)
Lag 5	—	—	-0.024 ($2.31e^{-5}$)
Lag 6	—	—	0.041 ($1.43e^{-14}$)
Lag 7	—	—	0.032 ($< 2e^{-16}$)
Lag 8	—	—	0.000 (0.930)
Lag 9	—	—	-0.028 ($< 2e^{-16}$)
Lag 10	—	—	0.000 (0.998)
Lag 11	—	—	0.017 ($3.09e^{-6}$)
Lag 12	—	—	-0.039 ($4.70e^{-14}$)
Lag 13	—	—	0.027 ($1.01e^{-7}$)
Lag 14	—	—	0.008 (0.102)
Lag 15	—	—	-0.027 ($4.90e^{-8}$)
Lag 16	—	—	0.013 (0.011)
Lag 17	—	—	0.017 ($1.27e^{-4}$)
Lag 18	—	—	0.005 (0.114)
Lag 19	—	—	0.000 (0.989)
Lag 20	—	—	-0.018 ($1.12e^{-9}$)
Lag 21	—	—	-0.006 (0.059)
Lag 22	—	—	0.000 (0.930)
Lag 23	—	—	0.047 ($< 2e^{-16}$)
Lag 24	—	—	0.197 ($< 2e^{-16}$)
Smooth Term Significance: EDF, χ^2, p-value			
s(AP)	6.50, 5871.3, $< 2e^{-16}$	6.41, 2409.7, $< 2e^{-16}$	6.28, 1181.7, $< 2e^{-16}$
s(LW)	6.75, 141.1, $< 2e^{-16}$	5.45, 68.81, $< 2e^{-16}$	7.29, 183.3, $< 2e^{-16}$
s(NF)	8.77, 4316.1, $< 2e^{-16}$	8.27, 1562.4, $< 2e^{-16}$	8.46, 132.5, $< 2e^{-16}$
s(OW)	7.35, 2002.8, $< 2e^{-16}$	6.25, 819.4, $< 2e^{-16}$	7.03, 738.0, $< 2e^{-16}$
s(TP)	8.93, 912895.4, $< 2e^{-16}$	8.94, 33512.6, $< 2e^{-16}$	8.87, 4903.1, $< 2e^{-16}$
Model Performance			
Adjusted R^2	0.964	0.980	0.987
Deviance Explained (%)	89.0	87.7	89.1

Bibliography

Allaire, J., T. Kalinowski, D. Falbel, D. Eddelbuettel, Y. Tang, and N. Golding (2024). *tensorflow: R Interface to 'TensorFlow'*. https://CRAN.R-project.org/package=tensorflow.

Banas, J. and K. Utnik-Banas (2021). "Evaluating a seasonal autoregressive moving average model with an exogenous variable for short-term timber price forecasting". In: *Forest Policy and Economics* 131. DOI: 10.1016/j.forpol.2021.102564.

Box, G., G. Jenkins, G. Reinsel, and G. Ljung (2015). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.

Brooks, S., S. Freeman, J. Greenwood, R. King, and C. Mazzetta (2008). "Quantifying conservation concern – Bayesian statistics, birds and the red list". In: *BIOLOGICAL CONSERVATION* 141, pp. 1436–1441.

Brundrit, G (2009). *Global Climate Change and Adaptation: City of Cape Town Sea Level Rise Risk Assessment*. Available at: http://oceangeoff@iafrica.com. Cape Town, South Africa: econlogic.

Chai, T. and R. Draxler (2014). "Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? – Arguments Against Avoiding RMSE in the Literature". In: *Geoscientific Model Development* 7.3, pp. 1247–1250. DOI: 10.5194/gmd-7-1247-2014. URL: https://doi.org/10.5194/gmd-7-1247-2014.

Chang, C. and C. Lin (2002). "Training v-Support Vector Regression: Theory and Algorithms". In: *Neural Computation* 14.8, pp. 1959–1977.

Chen, T. and C. Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. DOI: 10.1145/2939672.2939785.

Cho, M., C. Kim, K. Jung, and H. Jung (2022). "Water Level Prediction Model Applying a Long Short-Term Memory (LSTM)–Gated Recurrent Unit (GRU) Method for Flood Prediction". In: *Water* 14, p. 2221. DOI: 10.3390/w14142221.

Damle, C. (2005). "Flood Forecasting Using Time Series Data Mining". In: *University of South Florida*.

Dominici, F., A. McDermott, S. Zeger, and J. Samet (2002). "On the Use of Generalized Additive Models in Time-Series Studies of Air Pollution and Health". In: *American Journal of Epidemiology* 156.3, pp. 193–203.

Dubos, V., I. Hani, T. Ouarda, and A. St-Hilaire (2022). "Short-term forecasting of spring freshet peak flow with the Generalized Additive model". In: *Journal of Hydrology* 612, p. 128089. DOI: 10.1016/j.jhydrol.2022.128089.

Fang, Z., S. Yang, C. Lv, S. An, and W. Wu (2022). "Application of a data-driven XGBoost model for the prediction of COVID-19 in the USA: a time-series study". In: *BMJ Open*. DOI: 10.1136/bmjopen-2021-056685.

Fernández, C. and M. Steel (1999). "Multivariate Student-t Regression Models: Pitfalls and Inference". In: *Biometrika* 86.1, pp. 153–167. URL: https://www.jstor.org/stable/2673544.

Greenwell, B. (2017). *pdp: An R Package for Constructing Partial Dependence Plots*. DOI: 10.32614/RJ-2017-016. URL: https://doi.org/10.32614/RJ-2017-016.

Han, G. and Y. Shi (2008). "Development of an Atlantic Canadian Coastal Water Level Neural Network Model". In: *Journal of Atmospheric and Oceanic Technology* 25.11, pp. 2117–2132. DOI: 10.1175/2008JTECHO569.1.

Harrison, P. and C. Stevens (1971). "A Bayesian Approach to Short-term Forecasting". In: *Journal of the Operational Research Society* 22.4, pp. 341–362. DOI: 10.1057/jors.1971.78.

Hastie, T. and R. Tibshirani (1986). "Generalized Additive Models". In: *Statistical Science* 1.3. Accessed: 13-11-2024, pp. 297–310. URL: https://www.jstor.org/stable/2245459.

Hochreiter, S. and J. Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Huang, W., C. Murray, N. Kraus, and J. Rosati (2003). "Development of a Regional Neural Network for Coastal Water Level Predictions". In: *Ocean Engineering* 30.17, pp. 2275–2295.

Hyndman, R. and G. Athanasopoulos (2018). *Forecasting: Principles and Practice*. 2nd. OTexts. URL: https://otexts.com/fpp2/.

Hyndman, R. and Y. Khandakar (2008). "Automatic time series forecasting: the forecast package for R". In: *Journal of Statistical Software* 27.3, pp. 1–22. DOI: 10.18637/jss.v027.i03.

Kalinowski, T., D. Falbel, J. Allaire, F. Chollet, Y. Tang, W. Van Der Bijl, M. Studer, and S. Keydana (2024). *keras: R Interface to 'Keras'*. https://CRAN.R-project.org/package=keras.

Largier, J., S. Munger, F. Shilling, and R. Roettger (2023). "Forecasting High Bay Water Levels that Result in Flooding and Highway Closure". In: *National Center for Sustainable Transportation Research Report, March 2023*.

Lee, W. and T. Resdi (2014). "Neural Network Approach to Coastal High and Low Water Level Prediction". In: ed. by R. Hassan, M. Yusoff, Z. Ismail, N. Amin, and M. Fadzil, pp. 237–247. DOI: 10.1007/978-981-4585-02-6_24.

Lindemann, B., T. Müller, H. Vietz, N. Jazdi, and M. Weyrich (2021). "A survey on long short-term memory networks for time series prediction". In: *Elsevier, Procedia CIRP*. DOI: 10.1016/j.procir.2021.03.088.

Magnan, A., M. Garschagen, J. Gattuso, J. Hay, N. Hilmi, E. Holland, F. Isla, G. Kofinas, I. Losada, J. Petzold, B. Ratter, T. Schuur, T. Tabe, and R. van de Wal

(2019). *Cross-Chapter Box 9: Integrative Cross-Chapter Box on Low-lying Islands and Coasts*.

Maspo, N., A. Harun, M. Goto, F. Cheros, N. Haron, and M. Nawi (2020). "Evaluation of Machine Learning approach in flood prediction scenarios and its input parameters: A systematic review". In: *IOP Conference Series: Earth and Environmental Science* 479.1. DOI: 10.1088/1755-1315/479/1/012038.

Mosavi, A., P. Ozturk, and K. Chau (2018). "Flood Prediction Using Machine Learning Models: Literature Review". In: *Water* 10.11, p. 1536. DOI: 10.3390/w10111536.

Munger, S. and J. Largier (2022). "Input data for short-term water level forecasting at 3 stations near HWY 37, Sonoma/Marin County, California". In: *Dryad*. DOI: 10.25338/B8WS8H.

Ogunmolu, O., X. Gu, S. Jiang, and N. Gans (2016). "Nonlinear Systems Identification Using Deep Dynamic Neural Networks". In: *CoRR* abs/1610.01439. DOI: 10.48550/arXiv.1610.01439.

Rahman, A., C. Charron, T. Ouarda, and F. Chebana (2018). "Development of regional flood frequency analysis techniques using generalized additive models for Australia". In: *Stochastic Environmental Research and Risk Assessment* 32.1, pp. 123–139. DOI: 10.1007/s00477-017-1384-1.

Reguero, B., I. Losada, P. Díaz-Simal, F. Méndez, and M. Beck (2015). "Effects of Climate Change on Exposure to Coastal Flooding in Latin America and the Caribbean". In: *PLOS ONE* 10.7. DOI: 10.1371/journal.pone.0133409.

Ribeiro, M. and L. Coelho (2019). "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series". In: *Applied Soft Computing*. DOI: 10.1016/j.asoc.2019.105837.

Ripley, B., B. Venables, D. Bates, K. Hornik, A. Gebhardt, and D. Firth (2023). *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. https://CRAN.R-project.org/package=MASS. R package version 7.3-60.

Sansom, D., T. Downs, and T. Saha (2002). "Evaluation of support vector machine based forecasting tool in electricity price forecasting for Australian national electricity market participants". In: *Australian Universities Power Engineering Conf. (AUPEC)*.

Sapankevych, N. and R. Sankar (2009). "Time Series Prediction Using Support Vector Machines: A Survey". In: *IEEE Computational Intelligence Magazine* 4.2, pp. 24–38. DOI: 10.1109/MCI.2009.932254.

Smola, A. and B. Schölkopf (2004). "A tutorial on support vector regression". In: *Statistics and Computing* 14.3, pp. 199–222. DOI: 10.1023/B:STCO.0000035301.49549.88.

Tay, F. and L. Cao (2001). "Application of support vector machines in financial time series forecasting". In: *Omega* 29, pp. 309–317.

Trafalis, T. and H. Ince (2000). "Support vector machine for regression and applications to financial forecasting". In: *Proc. IEEE-INNS-ENNS Int. Joint Conf. on Neural Networks 2000 (IJCNN 2000)* 6, pp. 348–353.

Trapletti, A. and K. Hornik (2023). *tseries: Time Series Analysis and Computational Finance*. https://CRAN.R-project.org/package=tseries. R package version 0.10-55.

Trust, Sonoma Land (2021). *Highway 37 Redesign - Current Initiatives*. https://sonomalandtrust.org/current-initiatives/highway-37-redesign/. Accessed: 2025-01-31.

Tsakiri, K., A. Marsellos, and S. Kapetanakis (2018). "Artificial Neural Network and Multiple Linear Regression for Flood Prediction in Mohawk River, New York". In: *Water* 10.1158.

Tyagi, S., S. Chandra, and G. Tyagi (2023). "Climate Change and its Impact on Sugarcane Production and Future Forecast in India: A Comparison Study of Univariate and Multivariate Time Series Models". In: *Sugar Tech* 25, pp. 1061–1069.

Vapnik, V. and C. Cortes (1995). "Support-Vector Networks". In: *Machine Learning* 20, pp. 273–297. DOI: 10.1023/A:1022627411411.

Wang, Y. and Y. Guo (2020). "Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost". In: *Journal of Communications* 21.3, pp. 205–221. ISSN: 1673-5447. DOI: 10.23919/JCC.2020.03.017.

Willmott, C. and K. Matsuura (2005). "Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance". In: *Climate Research* 30.1, pp. 79–82. DOI: 10.3354/cr030079. URL: https://doi.org/10.3354/cr030079.

Wood, S. (2011). *Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models*.

Yurttas, G. (2024). "Forecasting Water Levels in Twente Canal Using Time Series Analysis". In: *University of Twente*.

Zhu, L. and N. Laptev (2017). "Deep and Confident Prediction for Time Series at Uber". In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*.