Distribution Agreement

Signature:

Arshia Singhal March 30, 2022

Modeling and Predicting Storm Surges Using Machine Learning Methods

By

Arshia Singhal

Talea Mayo Ph.D.
Advisor

Department of Mathematics

Talea Mayo Ph.D.
Advisor

James Nagy, Ph.D.
Committee Member

Vilma Todri, Ph.D.
Committee Member

2022

Modeling and Predicting Storm Surges Using Machine Learning Methods

By

Arshia Singhal

Talea Mayo Ph.D.
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences of
Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Mathematics

2022

Abstract

Modeling and Predicting Storm Surges Using Machine Learning Methods

By Arshia Singhal

Machine learning methods offer significant potential to reduce computational cost in predictive modeling. Through a combination of unsupervised and supervised learning methods, key insights can be gleaned on past data and can be used to forecast future data. This is especially valuable in cases where the original model requires significant computational time and power, such as in the case of storm surge prediction. Although several such models exist to simulate and forecast storm surge, such as Sea, Land, and Overland Surges (SLOSH), Advanced Circulation (ADCIRC), and Delft3D, these models are often computationally expensive and time consuming due to the numerical techniques used to solve the associated partial differential equations. As a result, we aim to reduce this computational complexity by implementing k-means clustering, linear regression, decision tree, k-nearest neighbors, and artificial neural network machine learning techniques to predict storm surge based on storm characteristics.

Modeling and Predicting Storm Surges Using Machine Learning Methods

By

Arshia Singhal

Talea Mayo Ph.D.
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Mathematics

2022

Acknowledgments

I would like to thank my thesis advisor, Dr. Talea Mayo, for all of her help and support throughout the honors program. I would also like to thank Dr. James Nagy and Dr. Vilma Todri for being on my thesis committee and motivating me to pursue a project based on their course material. Finally, thank you to my family and friends for their constant support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Tropical Cyclones and Storm Surge

Tropical cyclones, such as tropical depressions, tropical storms, hurricanes, and typhoons, are some of the deadliest and costliest natural disasters both in the United States and globally. The World Meteorological Organization emphasizes such storms' destruction in its ranking of the 10 largest natural disasters in history with respect to deaths and economic losses: it finds tropical cyclones and floods to be the second and third leading causes of death and the first and second leading causes of economic loss around the world, respectively [1]. Together, storms and floods have resulted in 635,932 deaths and US$ 636 billion in losses over the 50-year period from 1970-2019. In the United States alone, storms accounted for eight of the top 10 disasters regarding economic losses, seven of which occurred within the past twelve years [1].

The copresence of storms and flooding as two of the most destructive natural disasters is not a coincidence. Due to their strong winds and heavy precipitation, tropical cyclones often cause a drastic rise in sea levels, resulting in significant storm surges and subsequent flooding especially in coastal cities. Climate change poses an added threat: global sea levels have risen by as much as eight inches over the past century and are expected to increase by one to four feet over the next century due to human-caused global warming [2]. Warmer ocean and sea surface temperatures are also likely to intensify storm wind speeds and precipitation levels, suggesting storms of greater intensity in future years. Together, these factors have already resulted in significant loss of life, as evidenced by the immense death tolls of the less recent Hurricane Flora, Hurricane Fifi, and Hurricane Galveston, and the more recent Hurricane Sandy, Hurricane Katrina, and Hurricane Maria. Evidently, storm surges and the resultant flooding are key causes of the destruction of life and property following storms. Being able to understand and predict storm surges is a crucial way to minimize this destruction in future years.

1.2 Hydrodynamic Modeling

Several efforts have been made to model storm surges based on different storm characteristics, specifically through hydrodynamic modeling. These models primarily use either statistical or numerical methods to model the storm surges. Statistical models use correlations between past and future forecasted storms to predict surge, yet the limited frequency with which hurricanes occur render statistical models difficult to derive due to lack of data. As a result, we focus primarily on numerical models. Numerical models solve a set of partial differential equations associated with physical shallow water models using numerical methods such as finite differences and finite elements.

Several shallow water equations, encompassing characteristics such as flow velocity, fluid column height, fluid density, and more are used in conjunction to create the hydrodynamic models. One of the most well known sets of such equations is the Navier-Stokes equations. These partial differential equations extend Euler's equations, which do not include a viscosity component, to model viscous Newtonian substances through the conservation of their mass and momentum [3]. The equations can be expressed for incompressible flows as

$$\begin{aligned} &\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\Delta\mathbf{u} && \text{(momentum)} \\ &\nabla\cdot\mathbf{u} = 0 && \text{(continuity),} \end{aligned}$$

where **u** contains the velocity components; t is the time; ρ is the density; μ is the inverse of the Reynolds number; and p is the pressure.

In the momentum equation above, the left hand side describes acceleration, while the first term on the right hand side describes pressure and the second describes viscosity. In the case where $\mu = 0$, there is no viscosity, resulting in Euler's equations.

Note that because we are modeling incompressible flows, the continuity equation requires constant density along the line of flow over time, which in turn means that the divergence of the velocity must be zero.

These two equations can be written in two dimensions as

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) = -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0,$$

where $\mathbf{u} = (u, v)$ contains the velocity components; (x, y) are the spatial coordinates; t is the time; ρ is the density; μ is the inverse of the Reynolds number; and p is the pressure. The two dimensions are modeled through depth averaging and used in fluid modeling applications.

Because these shallow water equations are complex and computationally difficult to solve, numerical methods aim to approximate their solutions. As previously mentioned, two examples of such numerical methods are finite differences and finite elements.

The finite differences approach approximates derivatives over the domain, which is discretized as an evenly spaced mesh grid. This results in a system of equations that can be numerically solved. Several variations exist, such as forward differences, backward differences, and centered differences. To illustrate an example, the centered difference approximations for the first and second derivatives of a function f at a point x are as follows, obtained by rearranging the terms of a Taylor's series expansion:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2},$$

with h a small number close to 0. Similar approaches can be used for the forward and backward approximations.

The finite elements approach extends the finite differences approach by breaking the domain into smaller parts, which are connected at nodes and which in conjunction comprise the mesh. Each element uses a number of polynomials to approximate the original function, creating a system of equations. Other properties of the element include loading conditions, material properties, nodal coordinates, and more, and together these can be used to numerically model and solve a partial differential equation. Since the finite differences approach is more applicable to this instance of storm surge modeling, the discussion on finite elements remains brief.

One example of a storm surge modeling effort is the National Weather Service's Sea, Lake, and Overland Surges (SLOSH) model. Developed in the 1990s from previously existing models, this model uses shallow water equations and estimates storm surge heights based on hurricane characteristics such as pressure, speed, size, and more [4]. A large number of organizations, such as the Federal Emergency Management Agency, the United States Army Corps of Engineers, and more, use it to evaluate storm surge threat and determine if certain areas need to be evacuated, rendering it a key tool in tropical cyclone work. Other storm surge models include the Advanced Circulation (ADCIRC) model, the Storm Surge Modeling System with Curvilinear-Grid Hydrodynamics in 3D (CH3D-SMSS), Delft3D, and more, which primarily differ based on their equations, numerical solution methods, and areas covered.

The data set we will soon consider uses the SLOSH diagnostic model to compute storm surge based on storm characteristics. In particular, the model uses a finite differences approach to solve the Navier-Stokes equations of motion over an area or basin, represented by a polar mesh grid as illustrated in Figure 1.1 [5].

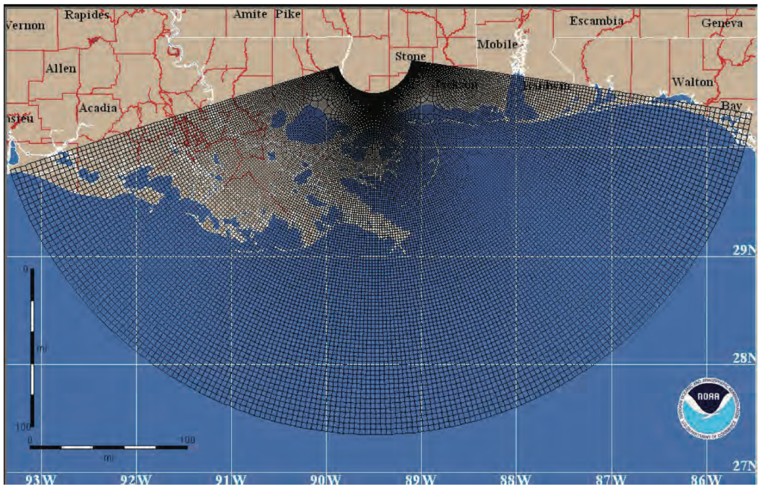


Figure 1.1: Polar Mesh Grid for the New Orleans Basin

These basins primarily cover coastal regions such as the U.S. Atlantic and Gulf of Mexico coastlines, and islands such as Puerto Rico and the Bahamas, which are all areas that experience greater exposure and susceptibility to hurricanes.

The SLOSH model can be run in three different ways to compute storm surge. The deterministic approach creates a single simulation based on meteorological input factors, thereby requiring accurate input for reliable predictions [4]. The probabilistic approach uses "statistics of past forecast performances" to create a group of SLOSH simulations that model astronomical tide [4]. The composite approach creates hypothetical hurricanes with different storm conditions by running the SLOSH model thousands of times, which creates Maximum Envelopes of Water (MEOWs) and the Maximum of MEOWs (MOMs). Of the three methods, the composite approach typically has the most comprehensive results since they account for forecast uncertainty through the multiple simulations [4]. Regardless of the approach used, the SLOSH model numerically solves a number of the previously described shallow water equations.

While these storm surge models are incredibly powerful, they are often computationally difficult and expensive to run. For the SLOSH model, deterministic predictions rely heavily on the accuracy of the input data and are often subject to significant errors. Because of this, the SLOSH model requires multiple simulations to accurately depict storm surge vulnerability, and is typically accurate within 20 percent under accurate input data [4]. The ADCIRC model has greater capabilities in varying its model's mesh resolutions than SLOSH but is less computationally efficient as a result [6]. In the case of an impending natural disaster, there is limited time to accurately gather the specific components and measurements to be able to run the models quickly and efficiently and subsequently predict the storm surge.

1.3 Background on Machine Learning Methods

Machine learning is a form of artificial intelligence that uses statistical methods and algorithms to make predictions based on a given data set. Through a combination of supervised and unsupervised learning techniques, machine learning methods can glean substantial information from a data set and extrapolate key observations and predictions. By applying these techniques to data on storms and their respective surges, this project aims to provide a faster and more streamlined approach to modeling and predicting storm surges.

Two primary machine learning categories include supervised and unsupervised learning techniques, distinguished by the absence or presence of a label, respectively. The label, also known as the target variable, is the independent variable, or the variable that we are aiming to predict or classify. In this case, the target variable is the storm surge.

1.4 Unsupervised Learning Methods

Unsupervised learning techniques are primarily used in the absence of a label in order to get an exploratory understanding of the data set. Some examples of supervised learning tasks include [7]:

1. Similarity Matching: used to identify similar data points.
2. Link Prediction: used to predict connections between data points.
3. Data Reduction: used to find a smaller data set that contains much of the information of the larger data set.
4. Clustering: used to group data points together by their similarity.
5. Co-occurrence Grouping: used to find associations between data points based on how often they occur together.
6. Profiling: used to characterize the typical behavior of individual data points or groups of data points.

Commonly used unsupervised learning techniques include k-means clustering, which groups similar data points together and organizes the data into several clusters, and the Hidden Markov Model, used specifically for modeling systems that are Markov processes. Besides unsupervised learning techniques, other exploratory analyses can be performed by observing the correlation matrix of the various variables and evaluating the data for outliers and immediately apparent patterns.

1.5 Supervised Learning Methods

In contrast to unsupervised learning techniques, supervised learning techniques are primarily used when the target variable is known. The goal is usually to classify or predict the target variable based on a set of features. Some examples of supervised learning tasks include:

1. Classification: used to determine to which of a set of classes the example belongs.

2. Regression: used to predict or estimate the numerical value of the target variable for the example.

3. Causal Modeling: used to understand what actions or events influence others.

4. Similarity Matching: used to identify similar data points.

5. Link Prediction: used to predict connections between data points.

6. Data Reduction: used to find a smaller data set that contains much of the information of the larger data set.

Notice that there is an overlap in certain tasks that can be performed using both unsupervised and supervised learning techniques, depending on the goal.

Several supervised learning methods exist, all of which have unique strengths and weaknesses that render them better suited for specific needs and purposes. Some of the most commonly used techniques include [8]:

1. Logistic regression: used for binary classification tasks and predicting the probability that an example falls in the positive class using a logistic sigmoid curve.

It may use maximum likelihood estimation as a measure of accuracy but might suffer from overfitting in the absence of feature selection. It is often better suited for smaller data sets, compared to decision trees.

2. Linear regression: similar to logistic regression, but for numeric predictions rather than exclusively binary classification. It may use empirical risk minimization as a measure of accuracy. Other extensions of regression include multiple regression and generalized linear models, which are more flexible generalizations of linear regression that allow for response variables that have error distributions other than a Gaussian distribution.

3. Decision tree: can be used for classification or numeric prediction tasks. It is often better suited for large data sets, compared to logistic regression, and can capture nonlinear relationships between the variables.

4. K-nearest neighbors (kNN): used for classification tasks but can be computationally expensive. It may suffer the curse of dimensionality in the case of many features. Similarity weighting can be implemented to make better use of a limited data set.

5. Bayesian inference: used primarily for classification tasks. It presents probabilistic models that can be used in various ways given the nature of the data and task. Naïve Bayes is commonly used for text classification and is efficient in both computation time and storage space required. Maximum a priori estimation can be used when there is information about the distribution of the parameters. Bayesian linear regression takes a full posterior distribution over the parameters into account when making predictions and computes the mean over all possible parameter settings, rather than fitting parameters or computing a point estimate of the parameters.

6. Support vector machine: similar to logistic regression in that it is primarily used for binary classification tasks. It estimates the probabilities of an example following into the positive or negative class and separates them using a hyperplane. It uses empirical risk minimization as a measure of accuracy through a constrained optimization problem.

7. Directed graphical models: used in probabilistic models to represent conditional dependencies. They allow hyperpriors to be placed on the parameters of the first layer of priors as a second layer of prior distributions and can be used to find the joint distribution's conditional independence relationship properties.

8. Autoregressive integrated moving average (ARIMA): used when data points are time dependent and seen as sequential rather than independent. It can be computationally expensive and is best suited for short-term forecasts.

9. Perceptron: used in classification tasks. This is a simple model that classifies an example into the positive or negative class by essentially finding a "line" that separates the data into the two class categories.

10. Artificial neural network: a more complex extension of a perceptron that is used for both classification and numeric prediction. It calculates the similarity between different neurons from the input layer and assigns them weights through a number of hidden layers, resulting in classifications or predictions in the output layer. It is best suited for complex models and can be an efficient way to find a computationally expensive solution.

1.6 Model Accuracy

Compared to unsupervised learning techniques, supervised methods are much more expansive and intricate. Because the methods aim to predict a value or classify it into a category, they use several parameters to measure and maximize the accuracy of these predictions. For instance, estimating parameters may require the use of maximum likelihood estimation (MLE) or maximum a posteriori estimation (MAP), for which the key computational problem is optimization. Similarly, Bayesian inference yields a posterior distribution with the key computational problem being integration. In the case of supervised machine learning methods, where the goal is empirical risk minimization, accuracy may be measured using a loss function, average loss, root mean squared error, or another metric.

An important concern with predictive models is to ensure that the model is not underfitting or overfitting the data. Regularization can help reduce the risk of overfitting by introducing a regularizer or a regularization parameter. Common methods to prevent overfitting include:

1. Ridge regression: use an L2-norm penalty in the form of the sum of the squares of the weights.

2. Lasso regression: use an L1-norm penalty in the form of the sum of the absolute values of the weights.

3. Feature selection:

   (a) Forward selection: add variables to the model, increasing its complexity until a stopping criterion is met.

   (b) Backward selection: remove variables from the model, decreasing its complexity until a stopping criterion is met.

4. Pruning (for decision trees): limit the number of branches and leaves based on a criterion.

Additional considerations are that some machine learning methods may require the assumption that the features are conditionally independent from each other, such as for Naïve Bayes, or that the examples are independent and identically distributed. Probabilistic principal component analysis can also be performed by using a latent variable z; this involves describing the data-generating process, simplifying the model structure, and defining simpler and richer model structures accordingly.

Once several machine learning models have been implemented, it is important to determine which performs best as a predictor. Holdout testing and cross validation can be used to better estimate the model's generalization performance across different data inputs. Holdout testing separates the data set into two parts: the training set and the test set. The training set is larger than the test set and is used to create the model. The test set is used to evaluate the performance of the model on unseen data. Cross validation, specifically k-fold cross validation, is the preferred extension of holdout testing since it evaluates the performance multiple times. Cross validation splits the data set into k equally-sized groups and takes one group to be the test data and the rest of the groups to be the training data. It then fits the model on the training data and evaluates the performance on the test group, as with holdout testing. The difference is that cross validation performs holdout testing k times such that each group is used exactly once as the test set, and then summarizes the performance across the k runs for a stronger generalization performance estimate than a single run of holdout testing would provide. Essentially, cross validation performs holdout testing multiple times to get a broader evaluation of the model's performance.

Nested holdout testing and nested cross validation can be used optimize the model's parameters and select which model works best by splitting the data into training, test, and validation sets. Different hyperparameter values are tested using the training set and their performance is measured on the validation set in an inner cross validation procedure. The optimal hyperparameters are then chosen based on performance outcomes in this inner procedure. Once the model's parameter values have been determined, the outer cross validation process uses the validation and test data to determine the generalization performance of the model. In effect, nested cross validation performs k-fold cross validation as described above, but it performs an additional cross validation on the training set of each fold to determine the optimal parameter value. Nested cross validation is a useful way to optimize model performance, but because it requires the model to be run and tested several times to test parameter values, it can be computationally expensive in the case of many parameters.

1.7 Applications to Storm Surge Modeling

Given the variety of machine learning techniques that are available, it is important to choose ones that are best suited for storm surge modeling purposes. Based on the different models' strengths and weaknesses and the given storm characteristics, we anticipate that the k-means clustering unsupervised method and the logistic regression, decision tree, k-NN, and artificial neural network supervised methods are most applicable for predicting storm surge.

Chapter 2

Methods

2.1 Data Curation

We analyze two key data sets, including “JamaicaBay_ncep_19801999_trk100.mat” and “surgedata.mat.” These data sets contain information on storm surges measured in and around Jamaica Bay, an estuary located in New York and close to New York City.

First, we process the “JamaicaBay_ncep_19801999_trk100.mat” data set to get a preliminary understanding of the data. The data set includes 29 features, the majority of which contain 100 time steps of data on 2000 pregenerated synthetic storms. These synthetic storms are created from a statistical-deterministic hurricane model, using the National Center for Environmental Protection/National Center for Atmospheric Research Reanalysis Project’s climatological data. A random seeding technique creates the initial hurricanes, and then the climatological data estimates the atmospheric state’s statistics and large-scale wind field. The wind field, along with “a coupled, deterministic atmospheric-ocean model,” is used to create synthetic hurricane tracks and simulate hurricane intensity [9]. Finally, the size of the hurricanes, measured as the outer radius of the hurricane and the radius of maximum winds, is defined using a lognormal distribution based on observational data [9]. The 100 time steps are equally spaced moments in time, each 10 minutes apart. From this data set, we extract six key features, namely:

1. pstore - pressure
2. rmw100 - radius of maximum wind
3. vmaxstore100 - maximum velocity
4. ro100 - outer radius
5. speed100 - translation speed
6. theta100 - translation direction

These will later be considered the independent predictor variables. We then determine the maximum (or minimum, for pressure) value of each of these six variables across the 100 different time steps and name them pstoremin, rmwmax, vmaxstoremax, romax, speedmax, and thetamax, as these would likely correspond to the maximum storm surge. We save these vectors of maximum values in a table so that the data can be easily accessible for modeling purposes. Note: ro100 is a constant value of 400 across all 100 time steps for all 2000 storms and will be excluded from further analysis.

We then process the "surgedata.mat" data set, which contains the data on the storm surge modeled using SLOSH at different times and areas. It contains three features, of which the primary focus is the "data" feature. This feature is a 2000x595x1517 matrix, containing storm surge values for 2000 storms over 595 time steps and 1517 locations. The 595 time steps are equally spaced moments in time within the same time interval as the first data set. The 1517 locations are taken from a grid of geographical points extending out from a central location in Jamaica Bay, according to the corresponding standard SLOSH basin [9]. As a first step, we calculate the maximum storm surge across all 595 time steps for each of the 2000 storms, at the location that the 70th location data point represents. We save these maxima in a vector called surgemax, which will represent the target variable in the supervised learning techniques.

2.2 Exploratory Data Analysis

We perform an exploratory analysis of the data set to get an understanding of the variables. The correlation matrix, created using the `corrcoef` command in MATLAB, provides a preliminary understanding of if any of the variables are strongly correlated. The correlation coefficients are listed in Table 2.1.

Attribute	**pstoremin**	**rmwmax**	**vmaxstoremax**	**speedmax**	**thetamax**	**surgemax**
pstoremin	1.000	0.457	-0.982	-0.175	0.111	0.044
rmwmax	0.457	1.000	-0.557	-0.150	0.089	0.040
vmaxstoremax	-0.982	-0.557	1.000	0.187	-0.117	-0.043
speedmax	-0.175	-0.150	0.187	1.000	0.003	-0.125
thetamax	0.111	0.089	-0.117	0.003	1.000	-0.092
surgemax	0.044	0.040	-0.043	-0.125	-0.092	1.000

Table 2.1: Variable Correlation Coefficients

From Table 2.1, it seems that maximum maximum velocity and minimum pressure are strongly inversely related, with a correlation coefficient of -0.982. None of the other variables seem to be very strongly related.

2.3 Unsupervised Machine Learning Methods: K-Means Clustering

The k-means clustering unsupervised learning method, performed on the five independent variables, determines the centroids of five different clusters and categorizes the 2000 data points into one of the five clusters. We use five clusters since there are five independent variables that we aim to differentiate, and we confirm sufficient variation by testing other cluster sizes and determining five to be informative. We use the `kmeans` command in MATLAB to perform the clustering and use squared Euclidean distance as the divergence measure with 10 max runs and 100 optimization steps in RapidMiner to better visualize the findings.

2.4 Supervised Machine Learning Methods

After completing the exploratory analysis, we use supervised learning methods to predict the storm surge target variable.

2.4.1 Linear Regression

We create an initial regression model using MATLAB's `fitlm` command, which uses ordinary least squares as its accuracy metric. The linear regression model uses the five independent variables of minimum pressure, maximum radius of maximum wind, maximum maximum velocity, maximum translation speed, and maximum translation direction to predict the maximum storm surge. It calculates the coefficient corresponding to each of the five variables, along with the intercept value.

Second, we create another linear model that only uses the three independent variables that were determined to be statistically insignificant in the first linear model to see if they have predictive value on their own. These variables are minimum pressure, maximum radius of maximum wind, and maximum maximum velocity.

Next, we try all of the 29 other possible subsets of the five independent variables as the predictor variables for storm surge by creating 29 additional regression models to see which combination of predictor variables yields the best performance.

Finally, we separate the surge data into two sets: one with extreme surge values and one without. We do so in two ways: in the first model, we split the surge data into values greater than 2 meters and values less than or equal to 2 meters; in the second model, we split the surge data into values greater than 1.34865 meters and values less than or equal to 1.34865 meters. The first threshold of 2 meters is chosen as being a level that causes a significant level of concern for coastal communities, while the second threshold is calculated by creating two k-means clusters from the maximum surge data and taking the average of the two centroids.

2.4.2 Decision Tree

We first create a decision tree in RapidMiner using the same data set as the rest of the techniques, with the five predictor variables and surgemax as the label. We use a relative split on the data set with a split ratio of 0.8 and shuffled sampling, with a least squares accuracy criterion and pruning applied.

We then create a second decision tree in RapidMiner where we treat the storm surge target variable as a categorical variable, with a surge of greater than two meters falling in the positive class and a surge of less than or equal to two meters falling in the negative class. This surge value was determined as an approximate level which would be a cause for concern to coastal cities. This change turns the numeric predictive task into a binary classification task.

2.4.3 K-Nearest Neighbors

First, we create the model in RapidMiner after normalizing the data and using 10-fold cross validation with shuffled sampling, k = 5, weighted vote, and Euclidean distance as the numerical measure.

Then, we introduce forward and backward selection in two additional models that have the same parameters as the first model to see if they result in a better predictive model.

2.4.4 Artificial Neural Network

We create the model in JMP using a randomized 2:1 training-to-test data split, a random seed of 1 for reproducibility purposes, and holdback validation. We use two hidden activation layers, the first with values of 3, 1, and 1 for the sigmoid tanH, identity linear, and radial Gaussian parameters, and the second with values of 2, 1, and 1 for the sigmoid tanH, identity linear, and radial Gaussian parameters, respectively. These values were determined to have some of the highest resulting R squared values for the test data.

Chapter 3

Results

3.1 K-Means Clustering

The centroids of the five clusters, summarized in Table 3.1, tend to be fairly similar in value but the clusters are of very different sizes, ranging from 206 elements to 779 elements. There is no particular distinguishing characteristic among the clusters. Both MATLAB and RapidMiner calculate the same centroids and cluster classifications.

Cluster	**pstoremin**	**rmwmax**	**vmaxstoremax**	**speedmax**	**thetamax**
0	963.886	53.372	55.759	18.187	358.957
1	966.543	60.501	50.974	17.019	168.871
2	922.438	54.790	80.427	18.523	358.979
3	998.167	112.528	21.627	15.035	356.854
4	988.648	64.759	34.130	15.937	358.609

Table 3.1: Cluster Centroids for Storm Characteristics

The heat map in Figure 3.1 illustrates whether certain clusters tend to have higher or lower values on average than the rest of the data set for any of the variables, excluding maximum speed. Features with values that are higher than average will appear green, and features with values that are lower than average will appear red. The lightness or darkness of the color indicates the magnitude of this difference, with darker reds and greens indicating a larger deviation from the mean. Note that cluster

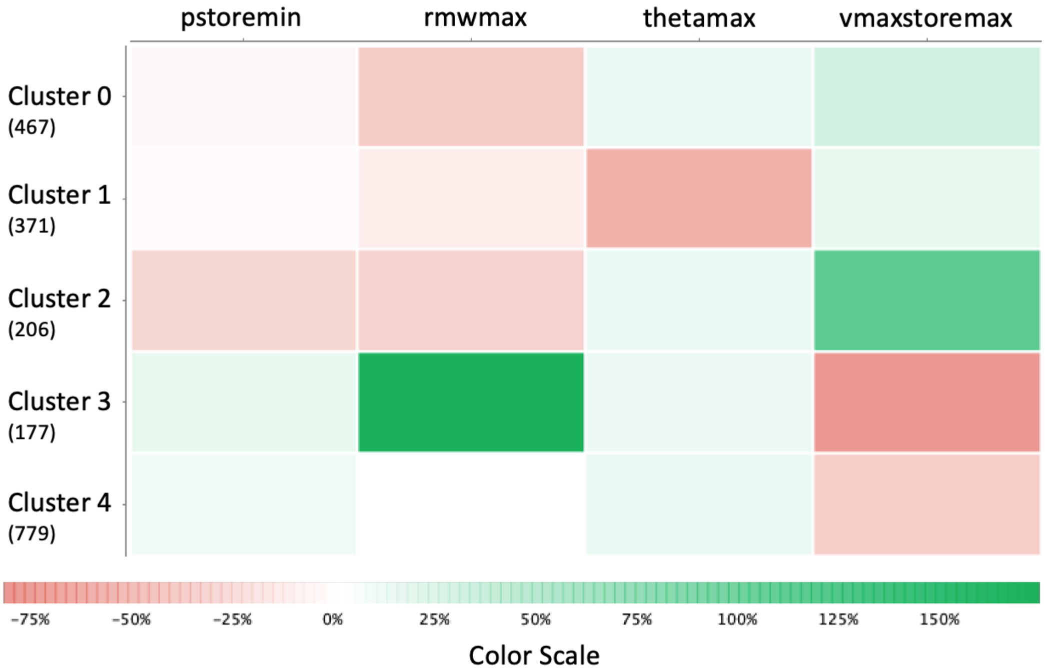


Figure 3.1: Cluster Centroids Heat Map

3 has a radius of maximum wind that is approximately 170% higher than the rest of the data, and a maximum maximum velocity that is approximately 80% lower than the rest of the data. This may suggest that for this cluster, maximum velocity and storm size, as determined by the radius of maximum wind, are distinguishing features that play a more significant role in determining storm surge as compared to the other variables.

The average and variance for storm surge is calculated for each of the five clusters in Table 3.2. Note that each cluster has a high variance in storm surge, so the surge values are dispersed rather than consistently being close to the average. Cluster 2 has the highest average storm surge, while cluster 3 has the lowest average storm surge. Recall that cluster 3's centroid had a high radius of maximum wind and low maximum maximum velocity; again, this may indicate a correlation with lower storm surge, or that radius of maximum wind and velocity play a larger role in determining the amount of storm surge. These characteristics will be explored further in a later analysis.

Cluster	**Average Storm Surge**	**Storm Surge Variance**
0	0.6905	0.5768
1	0.9123	0.7928
2	0.6547	0.5425
3	0.8397	0.8167
4	0.7263	0.6738

Table 3.2: Average Storm Surge For Each Cluster

3.2 Linear Regression

3.2.1 Using All Predictor Variables

The linear regression model outputs the coefficients associated with each of the five predictor variables along with the intercept, which it finds to be -4.6944. The model indicates that only two of the coefficients, namely those of max translation speed and max translation direction, are statistically significant, with p-values < 0.05. The other three variables have p-values > 0.05, indicating that the calculated coefficients are not accurate for the purposes of the model. The results seem to suggest that translation speed and direction are the best predictors of storm surge. However, the R squared value is only 0.0258, which is extremely low and suggests that storm surge is not being explained well by the predictor variables in this model. Once again, MATLAB and RapidMiner produce the same regression results since they use the same least squares method to calculate the coefficients, which are summarized in Table 3.3.

	Estimate	**SE**	**tStat**	**pValue**
1 (Intercept)	-4.6944	4.8145	-0.9751	0.3297
2 pstoremin	0.0058	0.0046	1.2591	0.2081
3 rmwmax	0.0018	0.0014	1.2864	0.1984
4 vmaxstoremax	0.00722	0.0066	1.0941	0.2741
5 speedmax	-0.0163	0.0031	-5.2267	1.9057e-07
6 thetamax	-0.0011	2.4762e-04	-4.2821	1.9393e-05

Table 3.3: Regression Output for First Regression Model

The root mean squared error (RMSE) for this model is 0.8140. Based on the results above, maximum surge should be predicted as follows:

$$maxsurge = -4.6944 + 0.0058 * p + 0.0018 * r + 0.0072 * v\text{–}0.0163 * s\text{–}0.0011 * t,$$

where p = pstoremin, r = rmwmax, v = vmaxstoremax, s = speedmax, t = thetamax. We can use this equation to calculate the predicted storm surge for each of the 2000 storms in the data set based on the five predictor variables and their values. We plot these predicted values against the actual values in Figure 3.2 to visualize the differences.

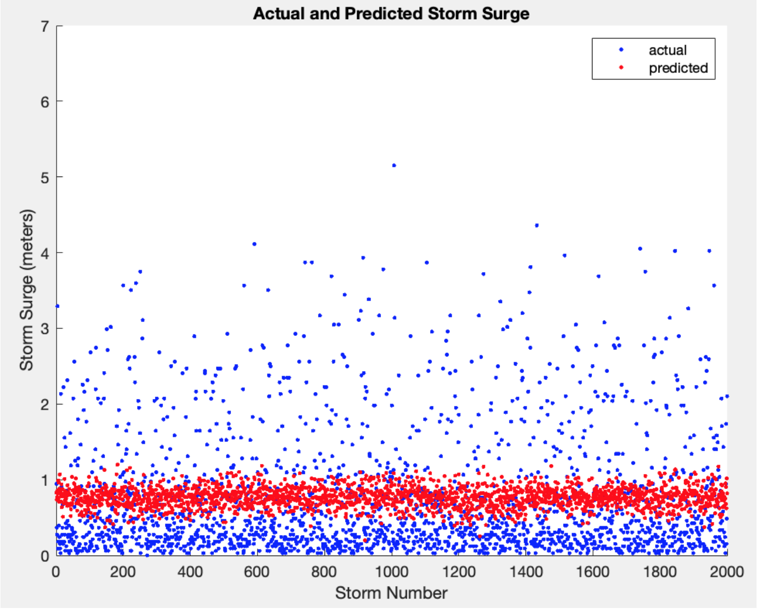


Figure 3.2: Actual vs. Predicted Storm Surge for First Linear Regression Model

Notice that the regression model's predictions are more concentrated than the actual surge values and tend to fall close to one meter in surge. The regression model also does not predict high surge values, such as surges of greater than 1.5 meters. This is not unexpected as most of the coefficients of the predictor variables were found to be statistically insignificant. However, we can create a threshold of tolerance around the predicted surge values, such as by saying that predictions within 0.5 meter of the actual surge are accurate. Doing so, we find that 69.85% of the predictions fall within this range, revealing the model to have greater validity provided slight flexibility with accuracy.

3.2.2 Using Subsets of Predictor Variables

Since only two of the coefficients were found to be statistically significant in the model above, we rerun the regression model on the three remaining variables, i.e. without maximum speed and maximum translation direction, to see if doing so might change the accuracy of the model. However, once again, as seen in Table 3.4, the three variables have high p-values that are greater than 0.05, rendering them statistically insignificant.

	Estimate	**SE**	**tStat**	**pValue**
1 (Intercept)	-4.9314	4.8666	-1.0133	0.3110
2 pstoremin	0.0054	0.0046	1.1682	0.2428
3 rmwmax	0.0019	0.0014	1.3684	0.1714
4 vmaxstoremax	0.0063	0.0066	0.9529	0.3408

Table 3.4: Regression Output for Second Regression Model

The R squared for this second regression model is 0.0029 and the RMSE is 0.8231. This suggests that the first regression model is slightly better, since it has a lower RMSE and higher R squared. Based on the coefficients of the second model, maximum surge should be predicted as follows:

$$maxsurge = -4.9314 + 0.0054 * p + 0.0019 * r + 0.0063 * v,$$

where p = pstoremin, r = rmwmax, and v = vmaxstoremax. Once again, we can use this equation to calculate the predicted storm surge for each of the 2000 storms in the data set based on the three predictor variables and their values. We plot these predicted values against the actual values in Figure 3.3 to visualize the differences.

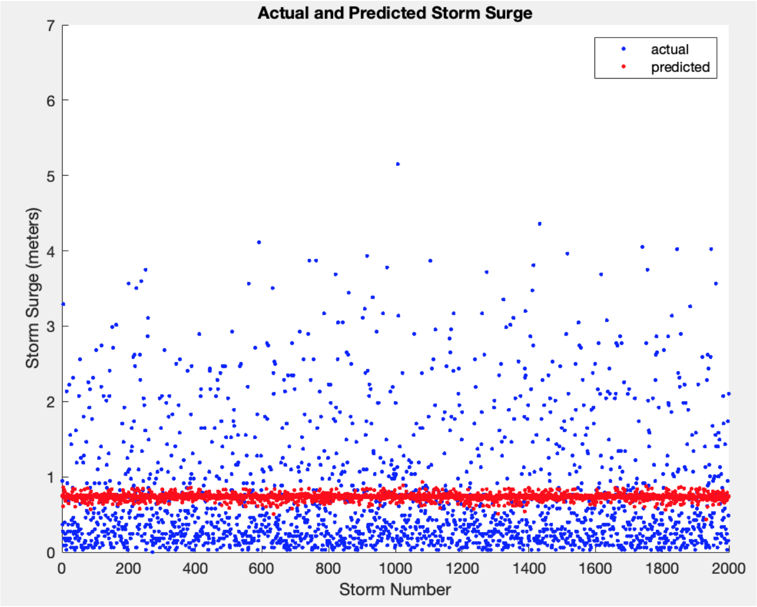


Figure 3.3: Actual vs. Predicted Storm Surge for Second Linear Regression Model

This second regression model's predictions are even more concentrated than the first model's predictions and the actual surge values, falling close to one meter in surge. The regression model also does not predict high surge values, such as surges of greater than one meter. This is not unexpected as the coefficients of the predictor variables were found to be statistically insignificant even after removing the two statistically significant variables. If we create a 0.5 meter threshold of tolerance around the actual values once again, we now find that 70.35% of predictions fall within this range, which is approximately the same proportion as the first model.

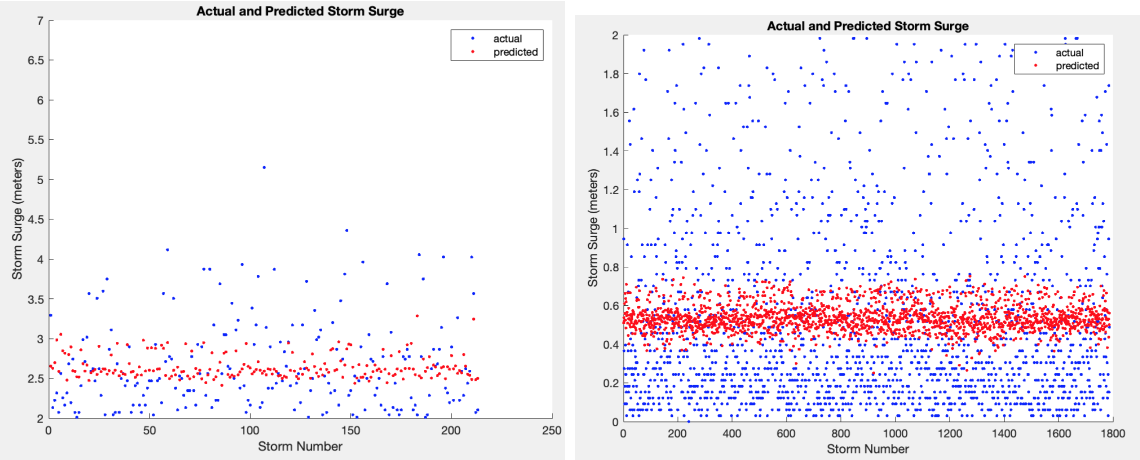
Finally, the other 29 linear regression models test all remaining possible combinations of the independent variables for one, two, three, and four predictor variable models. Across the 31 total models, the R squared value ranges from a low of 0.0016 for the model that only uses radius of maximum wind as the predictor variable, to a high of 0.0258 for the first model that uses all five variables. The RMSE ranges from a high of 0.8233 for the model that only uses radius of maximum wind as the predictor variable, to a low of 0.814 for the first model that uses all five variables. This suggests that for linear prediction models for this data set, using all of the predictor variables results in a more accurate model than eliminating any of the five independent variables.

3.2.3 Separating Extreme Surges

Because the first linear regression model is not predicting the high surge values, we consider splitting the data set into two to see if we have a better accuracy rate when we separate out the extremes.

Doing so for the first split model, with a threshold split at 2 meters, we get an R squared of 0.0579 for the maximum surge $\leq$ 2 meters data set and 0.0225 for the maximum surge $>$ 2 meters data set. 0.0579 is higher than the previous R squared values, which suggests an improvement when the extremes have been removed, as is intuitive. The RMSE also decreases from 0.8140 to 0.612 for maximum surge $\leq$ 2 meters and 0.463 for the maximum surge $>$ 2 meters data set, further suggesting an improvement. One interesting observation is that this extreme value data set now finds all of the predictor variables except radius of maximum wind to be statistically significant, while the data set without the extreme values finds only maximum translation direction to be statistically significant. The predicted versus actual values for the extreme surges data set and non-extreme surges data set are graphed in Figure 3.4a and Figure 3.4b, respectively.

For the 1.34865 meters threshold split model, we get an R squared of 0.0347 for the maximum surge $\leq$ 1.34865 meters data set and 0.0362 for the maximum surge $>$ 1.34865 meters data set. Again, 0.0347 is higher than the original R squared value, suggesting an improvement when the extremes have been removed, but it is not as high as the 2 meters threshold split model's R squared value. The RMSE also decreases from 0.8140 to 0.692 for maximum surge $\leq$ 1.34865 meters and 0.323 for the maximum surge $>$ 1.34865 meters data set, further suggesting an improvement. This threshold also finds all variables but radius of maximum wind to be statistically

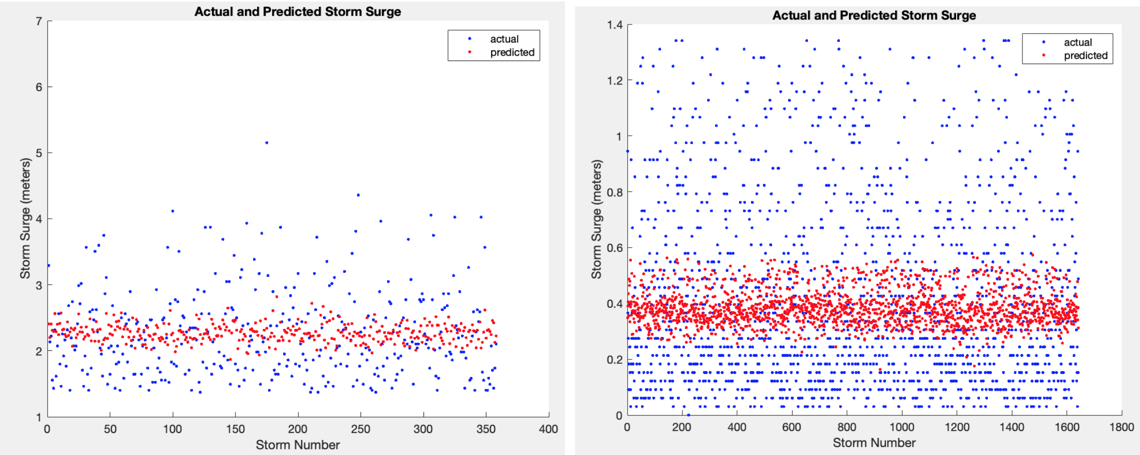


(a) Extreme Surges

(b) Non-Extreme Surges

Figure 3.4: Actual vs. Predicted Storm Surge for 2 Meter Split Linear Regression Model

significant for the extreme values and only maximum translation direction to be statistically significant for the non-extreme values, just like the 2 meters threshold split model. The predicted versus actual values for the extreme surges data set and non-extreme surges data set are graphed in Figure 3.5a and Figure 3.5b, respectively.



(a) Extreme Surges

(b) Non-Extreme Surges

Figure 3.5: Actual vs. Predicted Storm Surge for 1.34865 Meter Split Linear Regression Model

Despite the presence of statistically insignificant variables in these models, it may be that in the case of storm surge modeling, a variable's statistical significance is not necessarily the same as its importance to the prediction. P-values often border the threshold for statistical significance and p-values greater than 0.05, such as 0.1, are often also sufficient in some models. Thus, lack of statistical significance in these models should not render them invalid as they still reveal key insights into intervariable relationships. Ultimately, across the two split models, it seems that the lower split threshold allows for more accurate predictions for the extreme value data set but less accurate predictions for the non-extreme values data set, and is a small improvement on the original linear model without a data split.

One point to consider is the significance of the intercept in the case of storm surge. Typically, the intercept represents the value of the dependent variable when all of the independent variables are 0. In this case, this would mean that all five variables' maxima are 0, which means all of the data points across the original data set are 0. Because this is unrealistic in this physical scenario, it may be the case that the intercept does not play a meaningful role in storm surge prediction. We can rerun the first linear regression model where we set the intercept to 0 to see if eliminating improves the accuracy of the model. Doing so results in an RMSE of 0.8140, which is the same as the RMSE with the intercept, and an R squared of 0.0254, which is 0.0004 lower than the R squared with the intercept. Since neither the RMSE decreases nor the R squared increases, it seems that the absence or presence of an intercept does not play a significant role in the accuracy of the model in this case. Perhaps it relates to other weather factors or water levels that are not covered by the model, as discussed later.

3.3 Decision Tree

The first decision tree does not have a meaningful output due to the limited number of predictor variables and the numeric nature of the target variable. The decision tree only uses maximum speed and maximum translation direction as the predictor variables, which aligns with how the linear regression model determines these two to be the most informative attributes. This severely limits the predictive ability of the model so that it only produces three possible values for the storm surge based on speedmax and thetamax. This decision tree can be seen in Figure 3.6.

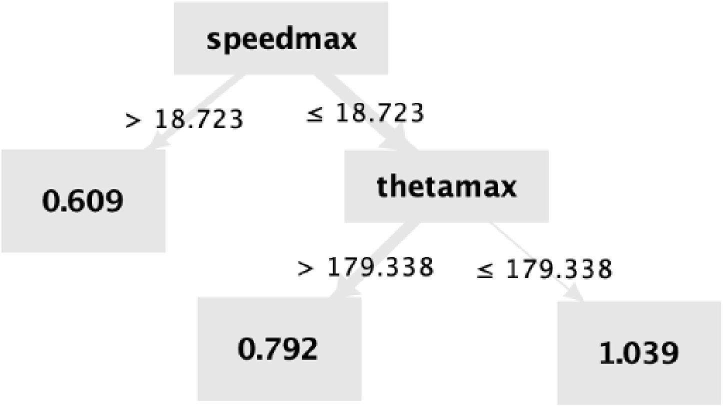


Figure 3.6: Numeric Prediction Decision Tree

The predicted versus actual values based on this decision tree are graphed in Figure 3.7. Notice that three possible values that the tree predicts are close to one meter, missing the extreme values as in past models.

The second decision tree, seen in Figure 3.8, has a different output since it has a binary classification task rather than numeric. Predicted surges that are greater than two meters are classified as 1, while predicted surges that are less than or equal to two meters are classified as 0. Once again, it only uses two predictor variables, which in this case are maximum maximum velocity and maximum speed. The tree classifies most of the predictions as 0, represented as blue in the leaf nodes. This is consistent with our previous models' findings of extreme surges being hard to predict. However, since most of the original surges are under 2 meters, the decision tree predicts 89.45% of them accurately, primarily missing the relatively few extreme surges.

The correct and incorrect predictions are illustrated in Figure 3.9, with red points

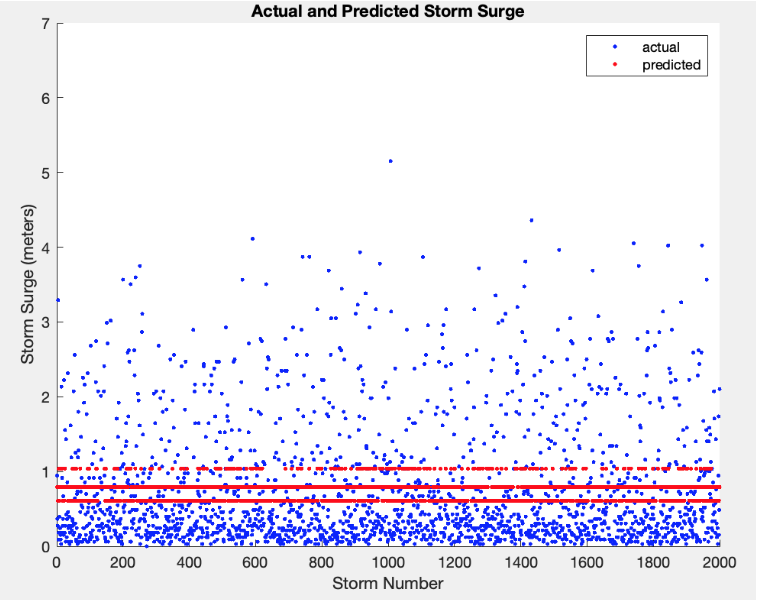


Figure 3.7: Actual vs. Predicted Storm Surge for Numeric Prediction Decision Tree

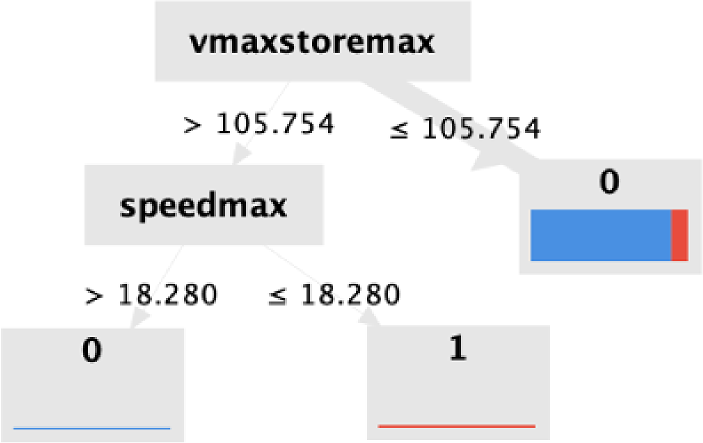


Figure 3.8: Binary Classification Decision Tree

representing incorrect predictions and green points representing correct predictions. Notice that all of the 0 values, or non-extreme surge values, are being predicted correctly, while only two of the 1 values, or extreme surge values, are being predicted correctly.

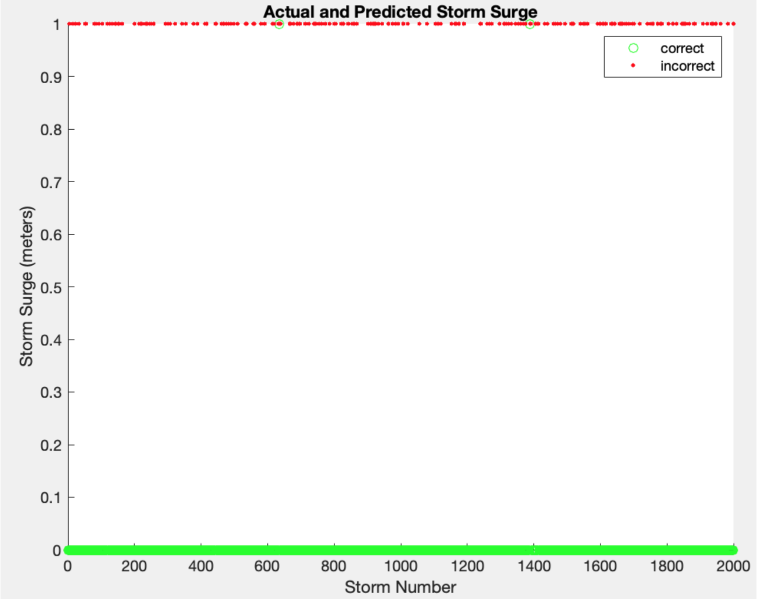


Figure 3.9: Actual vs. Predicted Storm Surge for Binary Classification Decision Tree

3.4 K-Nearest Neighbors

The first model, which does not have feature selection, has an RMSE of 0.883 and an absolute error of 0.651, which is fairly high given that the data set has been normalized.

The second model uses forward selection to aim to reduce the model's RMSE. It determines the minimum pressure, maximum speed, and maximum translation direction variables to be significant in improving the performance of the model and eliminates maximum radius of maximum wind and maximum maximum velocity. This model has an RMSE of 0.876, suggesting a slight improvement to the original model's predictive accuracy.

The third model uses backward selection to aim to reduce the model's RMSE. It determines the maximum radius of maximum wind, maximum maximum velocity, maximum speed, and maximum translation direction variables to be significant in improving the performance of the model and eliminates minimum pressure. This model has an RMSE of 0.877, suggesting a slight improvement in the model's predictive accuracy compared to the initial k-NN model but to a slightly lesser extent than the forward selection model.

Across the three k-NN models, the one using forward selection has the highest accuracy since it has the lowest RMSE, and suggests that pressure, translation speed, and translation direction are the most important factors in determining storm surge. However, the relatively large RMSE indicates that the model is ultimately not very accurate.

3.5 Artificial Neural Network

The resulting model has an R squared value of 0.037 for the training data and 0.036 for the test data. These values are similar to the R squared values of previous models and are quite low once again, even after trying different combinations of parameters to maximize them. The low R squared for the training data suggests that the model is underfitting the data, as overfitting would result in a high R squared for the training set and a low R squared for the test set. The training data has an RMSE of 0.815 while the test data has an RMSE of 0.795, which are also similar to those of past models. The neural network's input, hidden, and output layers can be visualized in Figure 3.10.

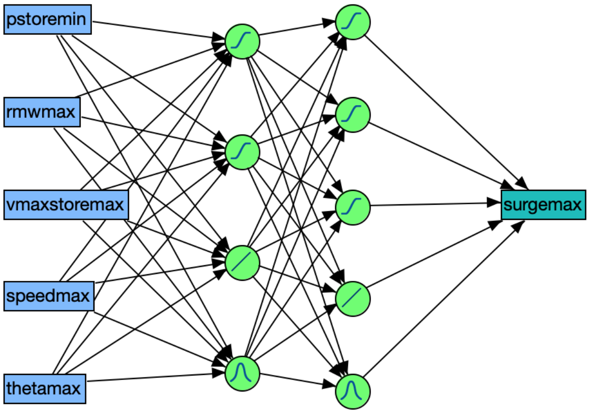


Figure 3.10: Artificial Neural Network Diagram

Graphing the model's predicted storm surges against the actual storm surge in Figure 3.11 reveals a pattern similar to that of the linear regression model. All of the predicted values for both the training and test sets are concentrated within a range of approximately 0-1.5 meters, indicating that higher surges are not being predicted accurately.

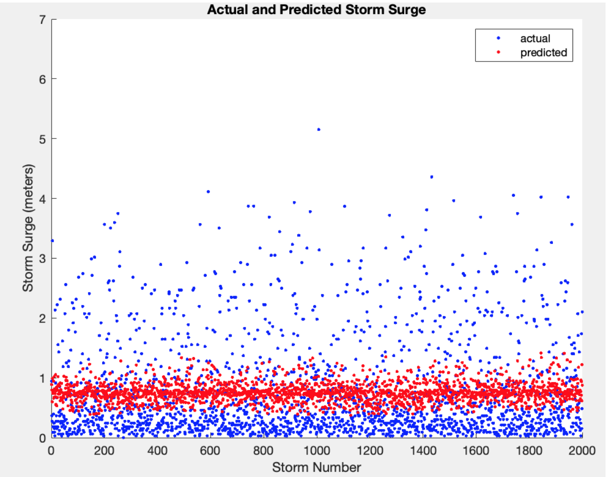


Figure 3.11: Actual vs. Predicted Storm Surge for Artificial Neural Network Model

Figure 3.12 further illustrates this insufficiency in predicting extremes. Points graphed closer to the diagonal black line would indicate more accurate predictions, yet most of the points appear away from the diagonal line.

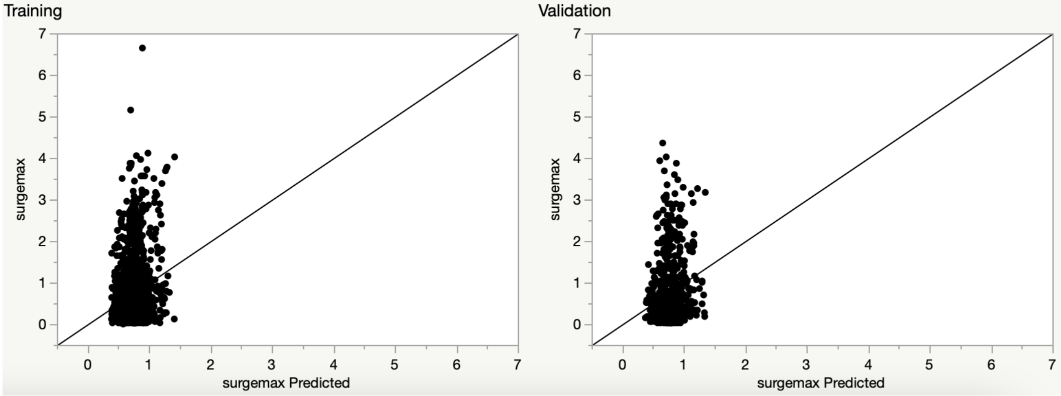


Figure 3.12: Actual vs. Predicted Storm Surge for Artificial Neural Network

Because the artificial neural network is a black box prediction model, it does not provide complete information on which predictor variables are most important to the output or how they are related to the prediction. It does reveal how heavily the nine different hidden layer nodes are weighted in their contribution to the output, along with the parameter estimates for the five independent variables within each node. However, little insight can be gleaned into how the variables relate to each other and how a change in the input would result in a change in the output.

Chapter 4

Discussion

Our results suggest that the data is insufficient for prediction using the linear regression, decision tree, k-nearest neighbors, and artificial neural net machine learning methods. The models created using these machine learning methods are not computationally accurate for the given data set and parameters. The R squared values are low across all of the models, indicating that the independent variables are not effectively explaining the variation in the storm surge dependent variable. However, they are slightly higher for the neural network than for the linear regression model, suggesting future potential for better prediction and insights with more robust data. The RMSE is also high for the models and several storm parameters are found to be statistically insignificant based on the high p-values, confirming the low applicability of the models.

Different models emphasize different parameters as being more important for predictive value. For instance, the initial exploratory analysis through clustering suggests that the radius of maximum wind and maximum maximum velocity may be more strongly correlated with storm surge. On the other hand, the first linear regression model finds the maximum translation speed and maximum translation direction to be the most statistically significant, as determined by their p-values being < 0.05.

The numeric prediction decision tree model agrees with the linear regression model in considering the maximum translation speed and maximum translation direction to be the most important variables, yet the binary classification decision tree emphasizes maximum maximum velocity rather than translation direction. The k-nearest neighbors forward selection model, deemed to be the most accurate of the k-nearest neighbors models, finds minimum pressure, maximum translation speed, and maximum translation direction to be the most significant predictors. These results are summarized in Table 4.1. The artificial neural network has been omitted since it is a black box model.

	Linear Regression	**Decision Tree**	**k-Nearest Neighbors**
pstoremin			X
rmwmax			
vmaxstoremax		X	
speedmax	X	X	X
thetamax	X	X	X

Table 4.1: Most Important Parameters for Each Predictive Model

Notice that all three models find the maximum translation speed and maximum translation direction to be two of the most important predictor variables.

One of the most significant reasons for the low accuracy of the models is their inability to predict large storm surge values. Recall that the predicted surges are concentrated in the 0-1.5 meters range in multiple models, rather than covering a range of 0-7 meters like the actual data. This suggests a lack of robustness in the models to be able to capture outliers and other less common values, since storms tend to have lower storm surges more often than higher surges. This is a significant limitation to the models; higher storm surges are typically more detrimental to communities impacted by tropical cyclones and hurricanes. Splitting the data such that extreme surge values are separate from non-extreme surge values and allowing a threshold of accuracy remedies this to some extent, but not to a significant level.

Although the machine learning methods used here do not predict the data well, past research has found success with certain methods, especially artificial neural networks. Researchers including Lee; Tseng et al.; De Oliveira et al.; Bajo and Umgiesser; Hashemi et al.; Kim et al.; Ayyad et al.; and more have created neural network models using different combinations of input parameters, hidden layers, and time intervals to successfully predict storm surge [10, 11, 12, 13, 14, 15, 16]. Ayyad et al. create an artificial neural network using synthetic storm data generated from the ADCIRC model to simulate storm surge [16]. Whereas SLOSH uses finite differences, ADCIRC uses finite elements to approximate and solve Reynolds-Averaged Navier Stokes partial differential equations, but most other elements of the modeling and analysis are similar. Ayyad et al. use more hidden layers and a larger number of nodes than past researchers to extract more information from the data and are ultimately able to create a model with an absolute percent error of less than 5% across all return periods, and a correlation coefficient between the predicted and actual data of roughly 0.97 [16]. Their success in using an artificial neural network to predict storm surge based on past characteristics suggests that with further data and exploration, machine learning methods may indeed be able to predict storm surge well.

Other limitations with the models in this project are based partly on the nature of the data. Currently, the independent variables include pressure, radius of maximum wind, maximum velocity, outer radius, translation speed, and translation direction. Incorporating other variables such as hourly water levels, wind direction, astronomical tide, and more into surge models and into the storm data could allow for a more robust and complex tropical cyclone models. In addition, the project currently only focuses on data from one location point, rather than modeling storm surge based on all of the location data. This may negatively impact the accuracy of the models given the large scope of the location and the geographical impact on storm surge. Next, the data uses the maximum or minimum value of each parameter for each storm across all of the time steps, rather than incorporating the time element. This significantly reduces the amount of data being utilized in the model, so reintroducing the time data in an alternative method to using the maximum or minimum might reveal more information into storm surge development over time. Finally, the data set is missing values for some time steps for variables such as maximum velocity and translation speed, which are simply replaced with a value of 0 or 1. These data points were not removed in this analysis due to the relatively small nature of the data set and the usage of the maxima or minima, but further analysis without missing data points may yield more accurate results as well.

Chapter 5

Conclusion

Machine learning methods offer significant potential to reduce computational cost in predictive modeling. Through a combination of unsupervised and supervised learning methods, key insights can be gleaned on existing data and can be used to forecast future data. This is especially valuable in cases where the original model requires significant computational time and power, such as in the case of storm surge prediction.

Although the models implemented in this project are not accurate in predicting the storm surge based on the five input parameters of pressure, radius of maximum wind, maximum velocity, outer radius, and translation speed, further analysis with a more robust data set may prove to be more successful, as past research has shown. Further research could include adding additional input parameters based on storm characteristics. Considering the impact of waves on top of surge and accounting for normal river flow and rain flooding could provide greater insights into the impact of increased water levels on coastal communities, though incorporating these elements would be complex and add an additional challenge. Analyzing the time series aspect of storm surge in how surge levels rise and fall across the time steps could reveal if different storm characteristics follow similar patterns over time, potentially increasing their predictive value. Exploring more nonlinear models could determine nonlinear relationships between the predictor variables and storm surge that the linear regression model fails to capture.

Finally, climate change poses an added level of complexity due to its effect on the physics of storms and storm surge. The current synthetic data is developed using patterns from historical events; while the physics of these events may not change in the future, the current data may not yet have observed all of the physical phenomena associated with future climate change, such as thermal expansion, changes in ocean current, and reduction in ocean salinity. As a result, data driven methods may not be the best for such applications, and methods that are constrained by physics may have stronger and more robust predictions.

Such physics-informed machine learning integrates physical laws with data in creating models and algorithms, such that predictions satisfy underlying physical principles [17]. This is especially helpful for smaller data sets where data-driven approaches may be less effective. Incorporating physical constraints allows the machine learning methods to remain more robust and accurate even if the data is imperfect, i.e. if it is noisy, has missing values, or contains outliers. These physical constraints can be incorporated in three ways: by introducing observational, inductive, or learning biases [17]. Introducing observational biases, considered to be the foundational and simplest incorporation of physics in machine learning, involves gathering or augmenting observational data such that it reflects the underlying physics. Inductive biases involve prior assumptions that guarantee that the predictions satisfy a set of physical laws, usually represented by mathematical constraints. Learning biases approximately satisfy the physical constraints by penalizing the loss functions and tuning the soft penalty constraints. Some examples of physics-involved machine learning methods include physics-informed neural networks, kernel methods such as Gaussian processes, and the deep Galerkin method [17]. These methods have been used to predict molecular properties, model turbulent fluid flow, enhance magnetic resonance imaging (MRI) data resolution, and more.

Despite the recent success of physics-informed machine learning, it also faces limitations in its approaches. Data acquisition costs can be prohibitively large when computed using large-scale computational models. Inductive biases require simple symmetry groups that are known a priori and may be hard to scale [17]. Some models, such as fully connected neural networks, have trouble learning high-frequency functions and fail to train as a result. Hybrid approaches aim to overcome some of these limitations by combining biases and their different elements. Regardless of current limitations, past research reveals that machine learning proves a promising tool in predicting storm surge, and using it in conjunction with physics constraints will make it even more powerful.

Bibliography

[1] World Meteorological Organization. Weather-related disasters increase over past 50 years, causing more damage but fewer deaths. https://public.wmo.int/en/media/press-release/weather-related-disasters-increase-over-past-50-years-causing-more-damage-fewer, August 2021.

[2] NASA. Sea level. Website, November 2021.

[3] Charles R Doering and John D Gibbon. *Applied analysis of the Navier-Stokes equations*. Number 12. Cambridge university press, 1995.

[4] National Hurricane Center and Central Pacific Hurricane Center. Slosh model. Website, January 2020.

[5] The NOAA Meteorological Development Laboratory. Slosh. Website.

[6] University of Rhode Island. Numerical models of storm surge, wave, and coastal flooding. Website, 2020.

[7] Foster Provost and Tom Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking*. " O'Reilly Media, Inc.", 2013.

[8] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.

[9] Talea L. Mayo and Ning Lin. Climate change impacts to the coastal flood hazard in the northeastern united states. *Weather and Climate Extremes, Under Revision*, 2022.

[10] Tsung-Lin Lee. Back-propagation neural network for the prediction of the short-term storm surge in taichung harbor, taiwan. *Engineering Applications of Artificial Intelligence*, 21(1):63–72, 2008.

[11] C.M. Tseng, C.D. Jan, J.S. Wang, and C.M. Wang. Application of artificial neural networks in typhoon surge forecasting. *Ocean Engineering*, 34(11):1757–1768, 2007.

[12] Marilia MF De Oliveira, Nelson Francisco F Ebecken, Jorge Luiz Fernandes De Oliveira, and Isimar de Azevedo Santos. Neural network model to predict a storm surge. *Journal of applied Meteorology and Climatology*, 48(1):143–155, 2009.

[13] Marco Bajo and Georg Umgiesser. Storm surge forecast through a combination of dynamic and neural network models. *Ocean Modelling*, 33(1):1–9, 2010.

[14] M Reza Hashemi, Malcolm L Spaulding, Alex Shaw, Hamed Farhadi, and Matt Lewis. An efficient artificial intelligence model for prediction of tropical storm surge. *Natural Hazards*, 82(1):471–491, 2016.

[15] Sooyoul Kim, Shunqi Pan, and Hajime Mase. Artificial neural network-based storm surge forecast model: Practical application to sakai minato, japan. *Applied Ocean Research*, 91:101871, 2019.

[16] Mahmoud Ayyad, Muhammad R Hajj, and T.-L. 2008. Back-propagation neural network for the prediction of the short-termstorm surge in Taichung harbor Taiwan. Eng. Appl. Artif. Intell. 21 63–72. Marsooli, RezaLee. Artificial intelligence

for hurricane storm surge hazard assessment. *Ocean Engineering*, 245:110435, 2022.

[17] Kevrekidis-I.G. Lu L. et al. Karniadakis, G.E. Physics-informed machine learning. *Nat Rev Phys 3, 422–440*, 2021.