The use of Ocean Buoy Data for Time Series and Machine Learning Forecasts

Wave Forecasting

Carlos Estuardo Amiel

2020-July-31

A Capstone Project Paper Submitted in Partial Fulfillment of the Requirements for the Degree Of

Master of Science

in Data Science

University of Wisconsin - La Crosse

La Crosse, Wisconsin

Abstract

Ocean buoy data is used in and of itself to develop forecasts of significant wave height using time series and machine learning techniques available in R: a language and environment for statistical computing. By using R, data can be collected from ocean buoys directly and used to produce reasonable multi step ahead forecasts. Southern California was used as the study region, specifically using data from the Santa Monica Bay and San Clemente Basin buoys. Models were evaluated using training and test data sets, with one model employing a more complex nested cross validation schema. Error measures were lower when forecasting data from the Santa Monica Bay buoy with one time series model and two separate machine learning models producing mean absolute error measures of .293, .226, and .259 meters respectively. These same models for the San Clemente Basin buoy produced mean absolute error measures of .477, .394, and .374 meters respectively. Ultimately, the methodologies used and workflows developed may prove to be a viable framework for data scientists who wish to study and forecast ocean waves.

Keywords: Ocean Buoys, Time Series, Machine Learning, Ocean waves, Open Source

Contents

List of Figures

List of Tables

Introduction

This case study will show how open source software coupled with data generated by the National Oceanic and Atmospheric Administration (NOAA) as well as the Coastal Data Information Program (CDIP), Integrative Oceanography Division, operated by the Scripps Institution of Oceanography, can be used to analyze and forecast aspects of ocean waves by Data Scientists who are interested in Oceanography or Ocean Engineering. Ocean observation data is used for various activities related to Climate, Coastal and Marine Hazards and Disasters, Ocean and Coastal Energy and Mineral Resources, Human Health, Ocean and Coastal Resources and Ecosystems, Marine Transportation, Water Resources, Coastal and Marine Weather, and Reference Measurement which are all deemed Societal Benefit Areas by the NOAA (National Data Buoy Center, 2018c). In addition to these broad categories ocean observations can help to initialize forecast models and simulations as well as to help verify their accuracy (Rohweder et al., 2012). From an economic standpoint the ocean economy has the potential to contribute up to US$1.5 trillion, with projections that between 2010 and 2030 this figure could double. What is clear from this data is that the ocean observations and the analytic activities it feeds are the anchor of the societal and economic value add of the ocean economy (Mackenzie et al., 2019).

In general, waves that occur in the ocean are most often caused by wind. Wind is generated by various types of weather systems.These weather systems can be characterized generally as areas of high and low air pressure where wind is generated as as result of the interaction between them. This interaction is areas of high air pressure flowing into areas of low air pressure which ultimately creates wind (Collins, 2020). As the wind blows over the surface of the ocean it creates disturbances on the ocean surface. Near the weather system these surface disturbances may be jumbled and have no apparent form and are termed “heavy sea” or seas. However, as waves slowly move away from the source they become more organized into what are called swell, which are characterized by a wave crest that is more so rounded and smooth compared to seas (Lovejoy, 2012).

There are three things that determine the size of an ocean wave when it is created by wind. The first is how fast the wind blows (speed), the second is how long the wind blows (duration), and the third has to do with the extent of water that the wind blows over (fetch). These three elements work together to generate waves that are a function of an increase or decrease in all three of the aforementioned factors i.e. fast wind speed, coupled with a large duration time over a vast expanse of open ocean will in general produce very large waves, whereas the opposite is true if all three elements were dialed down. In order to understand the dynamics of a wave it is helpful to see a picture of the various components. CDIP, Scripps Institution of Oceanography (2019) provides a simple yet effective picture that shows the important components of a wave. Wave height is the measure between the crest of a wave and the successive trough. The wavelength is the measured length between two successive wave crests. The wave period is the time in seconds that it takes for the wave crest to pass a point, meaning longer wavelengths correspond to higher periods. The importance of wave period is two fold: more energy and more interaction with the ocean floor as the wave moves from deep to shallow water.

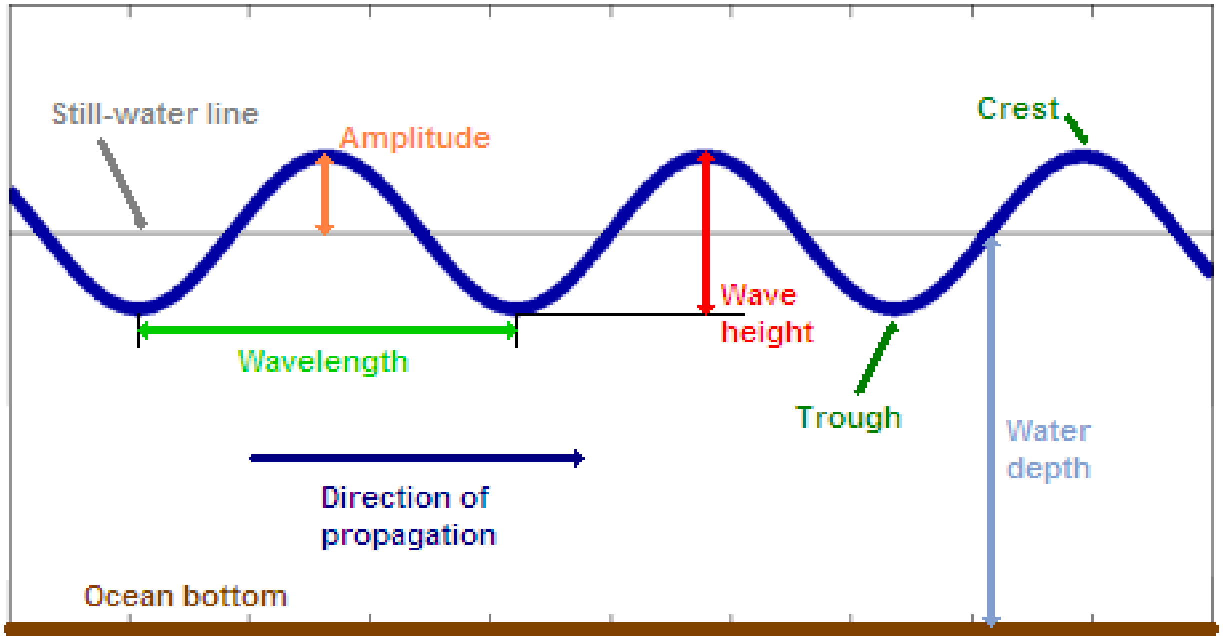

Figure 1: Wave Components by Coastal Data Information Program (2019)

The movement of waves affects water particles below the surface and this movement caused by the wave motion is called the wave orbit (Lovejoy, 2012). This is important because as waves move from deep to shallow water this movement, or wave orbit, interacts with the bathymetry of the ocean floor. For example waves with a higher period produce deeper below surface orbits which then behave differently as they make their way to shore. This interaction with the ocean floor is called shoaling. Refraction takes this idea of shoaling a bit further by describing the way a swell may bend toward shallow water as it interacts with the ocean floor. Ultimately swell period and ocean bathymetry work together to produce swell behavior unique to various combinations of both.

There are a number of sophisticated models that can model wind and waves, for example in Rohweder et al. (2012) a combination of ArcGIS, a Spatial Analyst License, Python and Pywin32 are software requirements. In addition the model requires inputs of adjusted wind speed, wind fetch, acceleration of gravity to determine wave height and wave period taking into account bathymetry to calculate maximum orbital wave velocity, while also taking into account friction factor and water density in the overall model. Another example is the SWAN wave model. This model is open source and accounts for wave-current interactions, spectral action balance equation, hydrodynamic flow, spherical co-ordinates or one dimensional balance equations, shoaling and refraction activity, wave generation by wind and various other considerations based on wave mathematical wave characteristics (Holthuijsen, 2010).

This study however will use the open source statistical programming software *R*, coupled with the freely available Integrated Development Environment (IDE) *R Studio.* Compared to the effort it takes to set up the aforementioned models and environments, installing and using R and R Studio is much easier. There are two main reasons for this. First, R is free and open source and therefore lends itself to more use by the academic community for develop cutting edge and up to date algorithms for a vast amount of topics spanning social sciences, physical sciences, math and statistics. Second, R boasts an active and friendly user community making it easier for a Data Scientist to find help for a specific topic and quickly iterate through any roadblocks.

The purpose of this study is to demonstrate, how a Data Scientist can use open source software (*R*) rich with sophisticated time series capabilities as well as cutting edge Machine Learning algorithms to engage with ocean observations with the intent of visualizing, analyzing and ultimately developing step ahead forecasts. These forecasts may then be used as benchmarks for various other models, or perhaps developed in and of themselves to improve understanding of ocean waves and their behavior with parsimonious models that may have a further reach than those developed and dedicated to ocean experts. In addition, as many activities associated with engaging in a big data environment are being abstracted away it is becoming easier to couple the power of *R* and open source programming with big data platforms that can handle large amounts of data. What this means is that this case study may be used as a starting point for analytic processes that can be deployed into big data environments and scaled to large data sets.

Different audiences may desire different ocean measurements, for example surfers most often desire long period swells because they increase the chance of producing larger breaking waves in shallow waters and therefore wave height may fall to the wayside. Wave height may be of greater importance to shipping vessels and others participating in open activities due to the surface disturbance they exhibit and the subsequent impact this may have. For this study we will focus our attention on the ocean observation measure wave height, specifically *significant wave height*.

The ocean observations in this case study are performed *in situ*, meaning an ocean instrument is either placed at or below the sea surface, for example a floating ocean buoy (Holthuijsen, 2010). The ocean buoy data that will be visualized, analyzed, and forecast in this study is *significant wave height* and is represented by the following equation that measures the mean of the highest one-third of all wave heights in a sampling period which spans 20 minutes for NDBC buoys (National Data Buoy Center, 2018b) and 30 minutes for CDIP buoys (CDIP, Scripps Institution of Oceanography, 2019).

$$H_{1/3} = \frac{1}{N/3} \sum_{j=1}^{N/3} H_j$$

In the equation above j represents the rank number of a wave given its height, with experiments showing this method of estimating wave height bears close resemblance to a visual observation and estimation of wave height (Holthuijsen, 2010). Two buoys and four years significant wave height data will be analyzed in an area of the pacific ocean specific to the southern California coast.It will be shown how data was collected and the various transformations performed on the data to prepare it for analysis. Subsequently the data will be visualized according to various time series focused methods and then used in various time series algorithms to develop forecasts. Finally the same data will be used in a machine learning framework that aims to use time based features to create step ahead forecasts.

Literature Review

In 1966 the Ocean Engineering Panel of the Interagency Committee of Oceanography recommended creating a national data buoy system. Shortly after, a feasibility study was done and Congress put in place legislation to support such a project. After the creation of the NOAA in 1970 the buoy program was transferred from the US Coast Guard to the NOAA with base operations located at the John C. Stennis Space Center in Mississippi where it still resides today and is known as the National Data Buoy Center (NDBC) (National Data Buoy Center, 2018c). The Coastal Weather Buoys (CWB) that are deployed by the NDBC are shown by the NOAA in National Data Buoy Center (2018a) to use the Geostationary Operational Environmental Satellite (GOES) system (Other systems may use Iridium satellites) as the first point of contact to receive data from a CWB where it then goes through a complex system of data transformation and quality checks before being made available to a wide variety consumers including the public.

Dr. Richard J. Seymor in 1975 with the assistance from the California Sea Grant Program began the Coastal Data Information Program (CDIP), which expanded quickly with funding from the California Division of Boating and Waterways and in 1977 funding from the U.S. Army Corps of Engineers (CDIP, Scripps Institution of Oceanography, 2019). What we see with the CDIP are *in situ* ocean measurement techniques that mirror those of the NOAA, where buoys are used to measure ocean characteristics such as wave height, wave period, and sea surface temperature and data is transmitted via satellite through a systems of checks before the data is made public.

While ocean waves can be produced by a variety of natural phenomenon such as underwater earthquakes and tidal forces the concentration here will be on waves caused in large part by forces of wind that disrupt the ocean surface. The idea is that as wind blows over the ocean, waves are in general a function of how strong the wind blows, how long it blows, and over how large of a distance the previous two factors occur (Collins, 2020). The field of ocean forecasting attempts to accurately model these causal factors as well as wave features themselves. Wave forecast models are then used to develop and guide various industrial and public safety applications as well as ocean engineering activities. Buoys that are deployed and maintained by both the NDBC and CDIP are instrumental in supporting the these activities and are the primary *in situ* ocean observation technique that will be in focus. A clear theme that emerges in a review of available literature related to ocean wave forecast models and buoys is that buoys are most often used as a verification system, or a check for how well physics based models perform. This is in stark contrast to the more recent modeling that takes into account solely the data that is captured and generated by ocean buoys.

Physics Based Models

Beginning in the early 1960s modeling ocean waves was in large part a statistics/physics based activity that was seeded in what is known as the *energy balance equation.* The equation is made up of two main components: physics based concepts that describe how waves behave and inputs from from physical processes such as wind, the effect of a wave breaking, as well as iterations that are nonlinear (Janssen, 2008). Even through it was generally understood that the *energy balance equation* as a whole did well in modeling ocean waves, much like a well established recipe that tells the user the exact quantities of each ingredient to use, it was not entirely known how much of each component of the energy balance equation contributed to the accuracy of the model.

Moving into the 1970s it became clear that there was an ingredient, so to speak, in the energy balance equation that was being overlooked in terms of how much it lent to modeling waves that were being generated close to the originating energy source. Here it's important to make the distinction between *windsea*, which are waves generated close to source i.e. wind, and *swell* which are waves that have moved away from the source. Primarily it was found that nonlinear interactions had played a large role in determining windsea characteristics far beyond what the wind was contributing. This evolution in wave modeling however was met with computing power that was unable to integrate these findings and therefore approximations were instead used (Janssen, 2008).

The 1980s ushered in a new wave of computing power, however it was still not realistic to expect exact modeling due to the complexity of nonlinear interactions. Janssen (2008) details how accurate parameterization of nonlinear interactions were instead developed and integrated into was is known as the WAM or WAve Model. These new calculations were integrated into new models such as Wavewatch III which emphasizes the wind wave relationship, and SWAN which in addition takes into account how waves interact with bathymetric features near shore. There is a clear evolution of wave modeling where new models put emphasis on using a new approach to a component of the energy balance equation and in some cases introducing interaction components to help describe not only windsea and swell but also near shore wave behavior.

Model Verification

Most often buoys are not used as a primary input into ocean wave model forecasting systems. In a study of the European Centre for Medium-Range Weather Forecasts (ECMWF) Janssen et al. (1997) illustrates the WAM model which features wind inputs, wave dissipation, energy transfer, and satellite generated altimeter data, with buoy data purposely not used in any model inputs but rather regarded as an independent verification of predicted wave heights. By combining high quality atmospheric wind modeling with satellite measurements the model performed well. In Bidlot et al. (2002) the WAM model is again shown to operate at a high level given high quality wind force data although a new focus was placed on sharing model results among scientific teams again with wind and wave data gathered from buoys acting as the backbone for model verification. Models, as shown in Behrens et al. (2018), may tend to under perform in extreme weather events such as those experienced in hurricanes and highlight the need for *in situ* ocean observations, in this case the CDIP network of buoys, to help increase understanding of wave variability beyond the models.

Model Initializing

O'Reilly et al. (2016) makes note that integrating buoy data into wind and physics based modeling is actively being researched, it is also noted that for the most part ocean engineering activities use buoy data for validation, however in this study we see that coupled with wind-wave physics models, buoy data is also used to initialize a wave modeling. A wave propagation model in Ludka et al. (2019) used the CDIP buoy network to estimate swell, sea waves and coastal waves. Similarly Crosby et al. (2019) uses buoy data to initialize models that are less computationally expensive than high resolution physics models. With these examples there is a move toward a hybrid model of sorts that not only uses physics based modeling dependent on high quality wind inputs but rather modeling that takes advantage of *in situ* observations from buoys to initialize models. Given the accessible nature of buoy networks such as those provided by the CDIP and NDBC there is a clear benefit to using these data sources for modeling activities.

Machine Learning

Machine learning is a topic in data science that can be separated generally into two camps: supervised and unsupervised machine learning. With supervised machine learning we train an algorithm to identify patterns based on data that is provided with correct and incorrect answers, so to speak. In this way the machine learns from the data what is a right or wrong answer either as a binary or continuous output. Unsupervised machine learning rather looks to find patterns in the data based on the characteristics of the data itself with no apparent right or wrong answer (James et al., 2013). This framework is opposed to a physics based model that uses the *energy balance equation* to simulate wave behavior under various conditions.

In a study by James et al. (2018) a machine learning framework to modeling wave conditions is put forth that takes as input, data simulated by the proven physics based SWAN model and is trained using both multi-layer perceptron (MLP) and support vector machine learning (SVM) algorithms to in effect output accurate wave conditions. A common theme with physics based models is that they are computationally expensive, yet what James et al. (2018) has shown is that a machine learning framework can operate thousands of times faster and therefore, in relation to the type of hardware needed to run a physics based model as opposed to a machine learning model, they are in effect more portable.

Other approaches to modeling significant wave height using a machine learning frame work have been proposed, for example in Mahjoobi and Etemad-Shahidi (2008) wind speed and direction are used to train artificial neural network (ANN) and decision tree algorithms to correctly output significant wave heights. Additionally even more research has been done in Mandal and Prabaharan (2006) exploring variations of ANNs such as Nonlinear Autoregressive with exogenous inputs (NARX) that improve on ANN performance.

What recent studies show is that a machine learning framework coupled with the wealth of ocean data being generated by high quality sensors *in situ* proves to be a worthy addition if not alternative or as James et al. (2018) puts it, a "surrogate" model approach to forecasting ocean waves. Data science as a discipline has been enabled by the robust nature of the current open source ecosystem. Both *Python* & *R* have emerged as pillars for developing machine learning platforms that are both cutting edge and scalable. In this case study the goal will be to showcase an end to end study from data acquisition to model output. Data will be sourced from ocean buoys, transformed, visualized, modeled, and forecast using only temporal and significant wave height input variables to produce step ahead ocean wave forecasts within the Southern California region.

Data Collection/Methodology

Methodological Approach

For this case study the idea was to show how data science principles can be applied to ocean wave forecasting. Specifically, how open source computing ecosystems like *R*, can help facilitate reproducible research, visualization, forecasting and ultimately lay the foundation for a data scientist to engage with ocean data. Ocean wave forecasting in this case study is characterized by extrapolating a time series for the purpose of providing insight into the future behavior of oceans waves. Quantitative data was needed in order to develop an appropriate time series data set and this data was found to be available as a secondary source collected *in situ* using ocean buoys. This data is curated by the NOAA and CDIP, each respective owners of ocean buoys that are able to provide various meteorological measures including our main interest: *significant wave height*.

Although ocean waves can be described using physics based models, these approaches are highly specialized and require a high level of oceanography domain expertise to understand and use. Here we are putting much less emphasis on the theory behind the *energy balance equation* and the influence it has on wave creation and rather take advantage of secondary data generated by ocean buoys and instead use these direct *in situ* measurements as the basis of our analysis of ocean waves. With this approach we can take advantage of a data science methodology put forth by Wickham and Grolemund (2016) that allows us to take a multifaceted approach to our data using the *R* statistical programming ecosystem, which in contrast to physics based models is well within the domain expertise of a data scientist who is interested in ocean data.

Data Collection Methods

The NOAA is a well known source of oceanographic data and provides many resources for various focus areas including ocean buoys. The NDBC website provides a visual representation of where buoys are located within the worlds oceans. This feature was helpful in identifying and selecting ocean buoys for the case study target area: Southern California. A user can zoom in on a target area, hover over icons that represent ocean buoys and retrieve relevant information about the selected buoy e.g. buoy ID, latitude and longitude location, what type of data it collected, data definitions, and how much historical data is available. The choice of which buoy(s) to choose from was based largely in part on how much historical data each buoy had available as well as if it collected a relatively uninterrupted stream of data for the selected measure: *significant wave height.* The NDBC web portal notes that *significant wave height* isn't directly measured but rather temporal sensor data that is Fast Fourier Transformed and further processed to account for any aberrations on account of the buoy structure itself. This transformed data are then ultimately the basis for calculating *significant wave height* as well as other measures such as average wave period.

Using the NDBC online portal it was determined that both the San Clemente Basin and Santa Monica Bay buoys fit the criteria of being located in the Southern California region, containing historical data from 2015 through 2018, and specifically data about *significant wave height.* Four years were chosen so that seasonal patterns, if apparent, could be seen at the highest grain i.e. years. Two buoys were chosen rather than one to help validate any inherent qualities that a time series of *significant wave height* may display. For example if a pattern is seen in the data of one buoy, would it also show up in the data of a nearby buoy. Both buoys report *significant wave height* in meters, and while data is stored in hourly increments, for this case study hourly measures were aggregated into daily averages.

The NDBC web portal provides "click and download" procedures for files containing relevant data, however this type of data retrieval does not lend itself to basis of the case study, which grounds itself instead on a programmatic approach to help facilitate reproducible research, specifically using *R*. In this light the *R* package `rnoaa` proved to be a significant resource and was used to retrieve data from the NDBC using a function designed specifically to query buoy data stores. The function takes in as parameters the type of data set (for this case study "standard meteorological" which includes significant wave height), the buoy id, as well the year for which the data was collected. This function was then used in a custom nested loop where the outer loop cycled through each buoy id and the inner loop through each year, which produced 4 years worth of hourly data for both San Clemente Basin & Santa Monica Bay buoys.

Study Area

In addition to using the `rnoaa` package to programatically retrieve data, the *R* ecosystem also contains `marmap`, an excellent package that helped to handle and visualize bathymetric data. Bathymetric data for the Southern California region was retrieved from the NOAA Grid Extract webpage then `marmap` was used to develop graphics that helped to visualize the study area. For example the map below was created using `marmap` and shows us, with color coded dots, the location of the buoys within the target region.

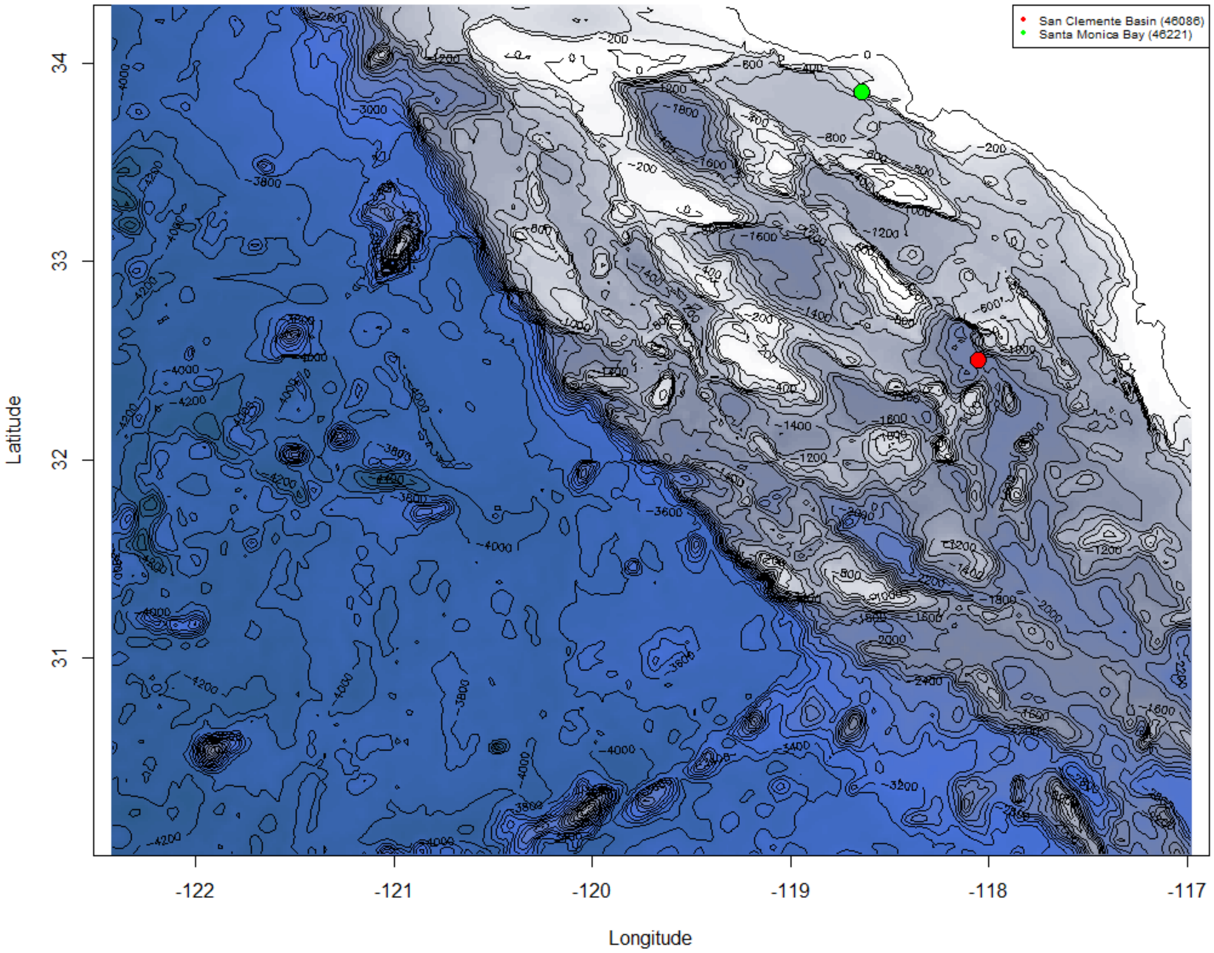

Figure 2: Buoy locations and bathymetry map

The bathymetry map of the region increased understanding of the underwater features of the study region as well as to visualize buoy placement in the region and their proximity to each other. The `marmap` package was also used to generate a transect view of the region which helped add relative context to the water depth below each buoy: 370 meters and 1845 meters for Santa Monica Bay and San Clemente Basin buoys respectively. An additional view of the study region bathymetry using a 3D plot was also generated using the `marmap` package. As a whole these visuals helped to visualize the physical environment where the buoys are located.

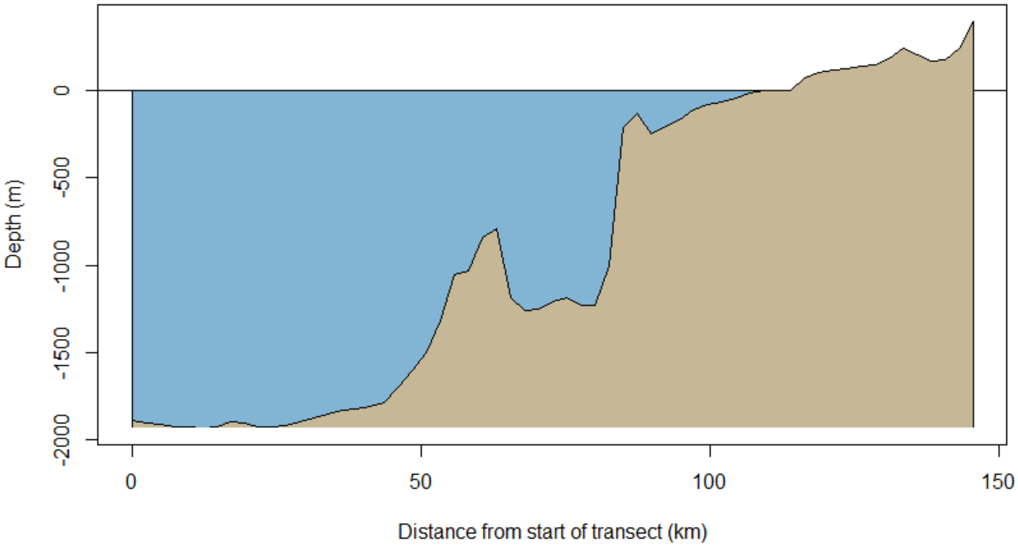

Figure 3: Study region transect map

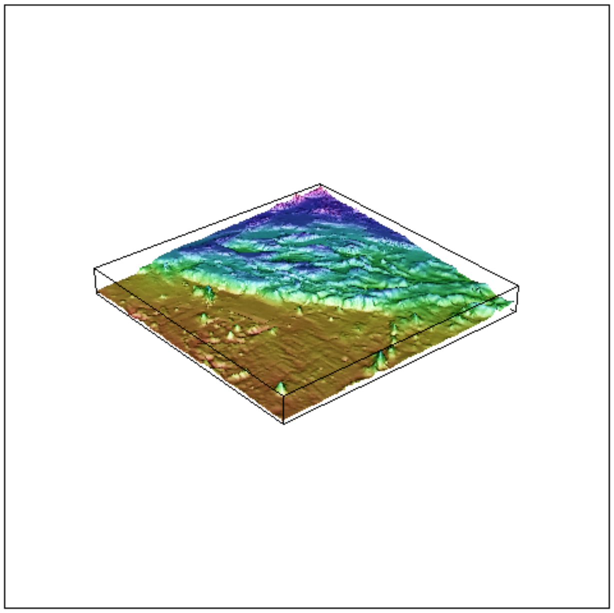

Figure 4: Study region 3D bathymetry map

Understanding the physical environment is important because it is known that these physical characteristics e.g. offshore islands and a diverse bathymetry contribute to the difficulty in forecasting ocean waves in this study region (Thomas et al., 2016). With this in mind, one of the central aims of the case study is to provide a purely *in situ* collected data based approached where buoys help frame up the forecasting challenges as a time series or supervised learning/machine learning problem.

Methods of Analysis

The `rnoaa` package was instrumental in developing a programmatic approach to retrieving relevant ocean buoy data from the study area. Within the code procedure to query and import buoy data from the NDBC, the data was formatted into into an `R` data structure known as a *data frame.* This data frame structure contained raw data for 18 variables retrieved from the "standard meteorological" data store, however for this case study only seven were considered and of these seven only two were considered as actively contributing to forecasting: *time* and *wave_height.* All seven variables can been seen in the table below. Note here that wave direction is measured as the direction waves are coming *from*, with north equivalent to 0, east to 90 and south to 180.

Table 1: Variables considered

measure	description
time	year-month-day hour:minute:second
buoy_id	buoy id
lat	latitude
lon	longitude
average_wpd	average wave period in seconds
mean_wave_dir	wave direction as degrees
wave_height	significant wave height in meters

Data Preparation

After the data was retrieved from the NDBC data stores, the first step was to make sure that the date index expressed as the *time* variable was in a format that enabled various time based transformations and visuals. This was done with tools loaded with `fpp3` package, which includes methods for creating and cleaning time based data tables. The *time* variable was initially imported as a string and was therefore converted into a date-time class that represents calendar dates and times. Once this step was completed it was possible to use the *time* variable to create time based visuals. Since time based data is inherently continuous, meaning there are no gaps in time, the next step was to identify any implicit time gaps. Below is a visual that displays the location of implicit time gaps in the hourly time series for both buoys.

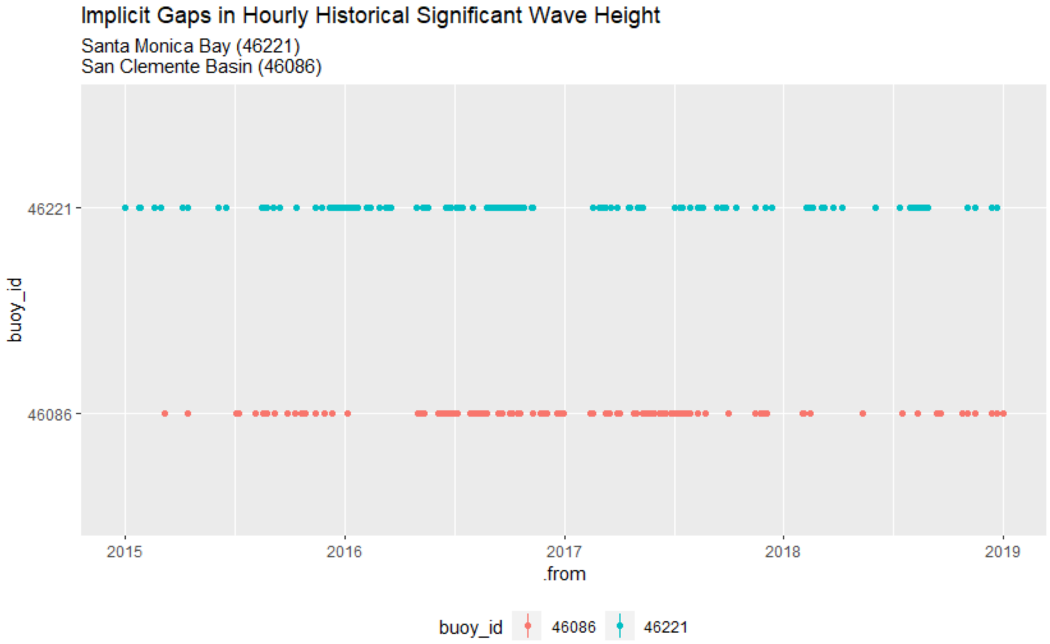

Figure 5: Implicit missing hourly values

These implicit missing values were then turned into explicit missing values, a step that also fills in these newly created explicit missing values with “NA”. In order to account for NA values the next step was to fill in these explicit gaps in time with values from the previous row. This is a simple method in contrast to others such as interpolation of values. However since the hourly data is unlikely to change dramatically from one hour to the next and given the scattered nature of the missing values this simpler method was employed. The last step was to aggregate the hourly data into daily values, which was done by computing the mean value of hourly data for each day. The resulting time series plot below shows the these derived daily values for both buoys and what was ultimately the base time series for the case study analysis of *significant wave height*.

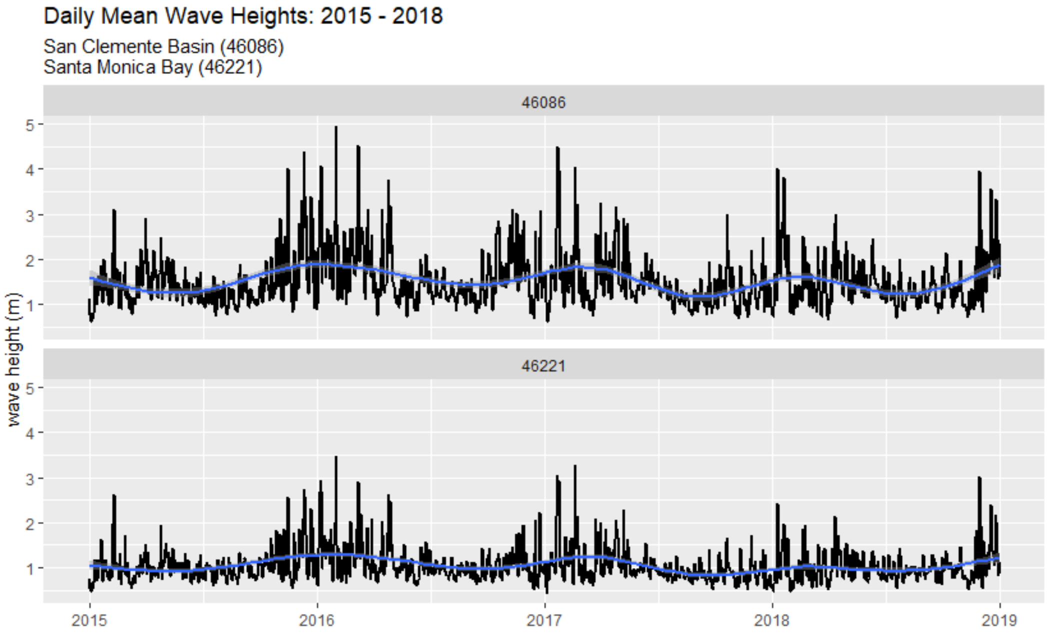

Figure 6: Daily mean wave heights

Time Based Visuals

Time based visuals can help to discern patterns in the data that aid in the overall analysis. For example in Hyndman and Athanasopoulos (2018) a proposed set of visuals are set forth that aim to quantify possible relationships between variables, seasonal components, or perhaps correlation between the target variable, here *significant wave height*, and a lagged versions of itself. These visuals were explored for this case study. As was shown in Venkatesan et al. (2019) it is entirely possible for a seasonal pattern to exist in a time series of buoy data, furthermore the relationship between variables can also bring to light regional tendencies. Similarly in this case study the relationship between *significant wave height* was compared to *average wave period* and *wave direction*. The plot below shows these relationships using the San Clemente Basin Buoy data as an example.

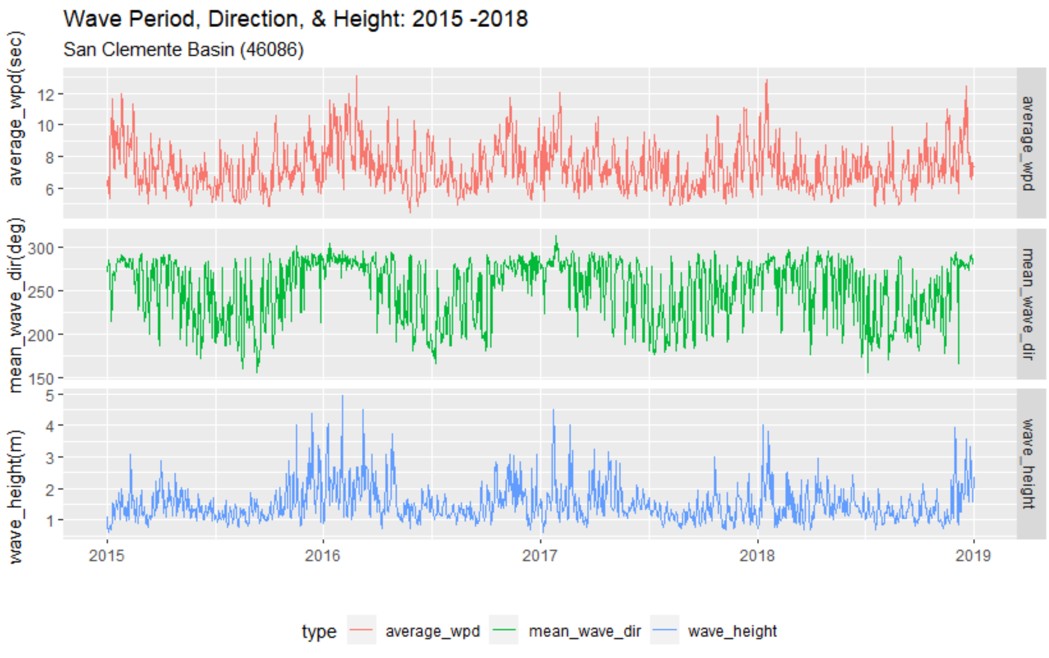

Figure 7: Wave height, period, direction

This seasonality was apparent for both buoys when the data was aggregated on the basis of month, although less pronounced for the Santa Monica Bay buoy.

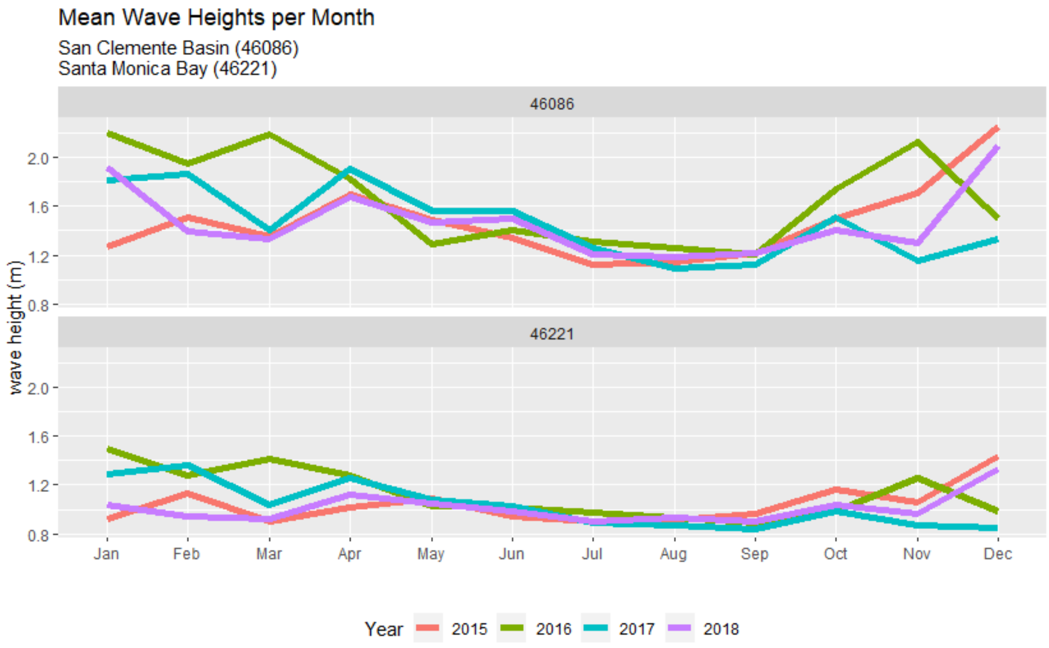

Figure 8: Daily mean wave heights month seasonality

What the plots above revealed is that the seasonal variations seen in the data i.e. elevated values of *significant wave height* early/late in the year with dips through the middle of the year, elevated values are accompanied by a northerly mean wave direction that generally exhibited long period swell signatures. This shows that as a general pattern, storms occur further north allowing for swell to build and travel a longer distance and may account for the larger wave heights as they arrive in the study region and are measured by the San Clemente Basin and Santa Monica Bay ocean buoys.

The presence of autocorrelation in the time series for both the San Clemente Basin and Santa Monica Bay buoys was apparent when visualized in a correlogram plot as shown below. As noted in Hyndman and Athanasopoulos (2018) the patterns in this visual helped show that there was no strong trend in the time series of *significant wave height* and values close in time are similar. In addition a monthly seasonal variation in the time series was apparent with the transition from a positive autocorrelation coefficient to a negative one between approximately 120 to 300 day lag times, with additional peaks and valleys in the first 90 and last 60 days of the year.

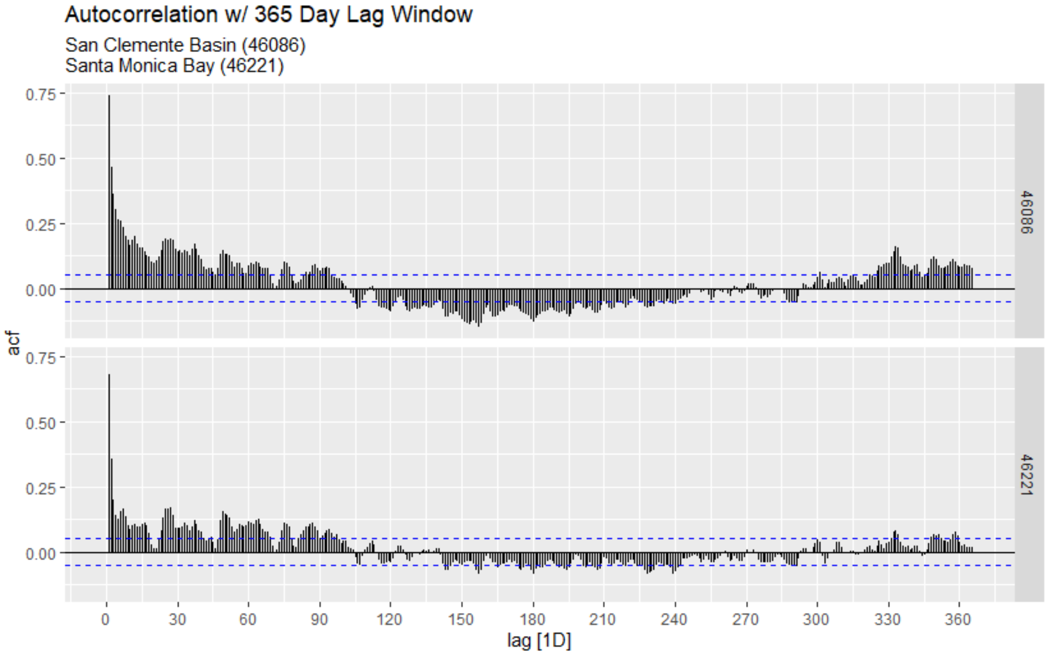

Figure 9: Autocorrelation 365 day window

The plot below helped to provide an additional view of the variation of wave heights for each month for each buoy. Each month showed a fairly normal distribution with only those months approximately in the first and last quarter displaying wider distributions, confirming the idea of monthly seasonality. Although the distribution reveals instances of high values in the aforementioned quarters none of these values were removed because as shown in Thomas et al. (2016) under prediction is apparent near shore in this region and any modeling might do well to account for these large values that may not represent anomalies but rather seasonal tendencies.

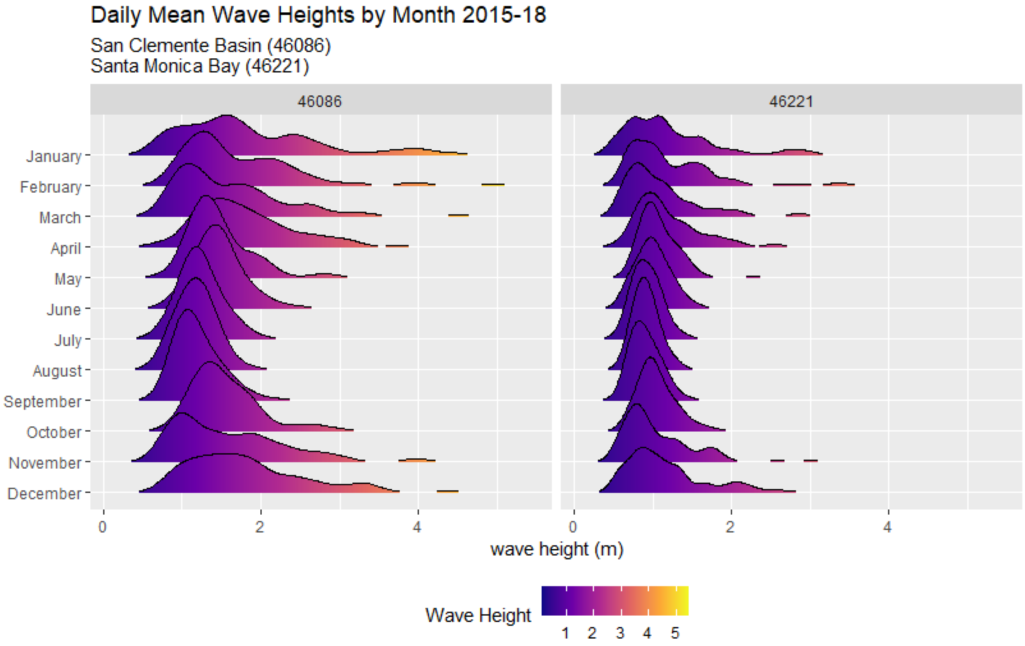

Figure 10: Daily wave height distribution by month

Analyzing the patterns in *significant wave height* for this case study helped to provide an overall idea of how the data behaved over time and what values could be expected when forecast models were built. The patterns in *significant wave height* identified by comparing it to other variables, aggregating across different time dimensions as well as analyzed in relation to lagged versions of itself, also helped to confirm the presence of statistically significant time based variations.

Time Series Models

Tools loaded with the `fpp3` package contain a suite of forecasting methods that were used for this case study. The time series used were univariate since only the *time* and *wave height* variables were considered. There were five forecasting methods used:

1. Exponential smoothing state space model (ETS)
2. Autoregressive Integrated Moving Average (ARIMA)
3. Seasonal naive (SN)
4. Neural Network (NN)
5. Ensemble of ETS, ARIMA, and SNAIVE (ES)

In order to assess model accuracy the time series for each buoy was split into train and testing regions. The training region was set to $n - 10$ days with n being the total number of days. This was done with the goal that each model would then generate a ten day forecast. The specialized time based training data frame containing the time series for both buoys and $n - 10$ training data was then passed into a model function. This function takes as arguments the training data, as well as all modeling functions that will be used e.g. ETS, ARIMA, SN, and NN. For each of the four models all model parameters were left to their defaults and each function was allowed to determine automatically the best parameters. The model function then takes the training data and iterates it through each identified model, effectively building four models with the fifth ensemble model representing an average of ETS, ARIMA, and SN. Each of these models were then passed to a forecast function which conveniently combines the 10 day forecast with the original set of data.

A visual inspection was done on the model and their fitted values to ensure that moving forward any modeling activity was moving in the right direction judged in part by how well the model fit the actual of the training data set. A graphical representation of each of the four trained models and their fitted values can be seen in Figure 11. Each of the models fit the training data set fairly well.

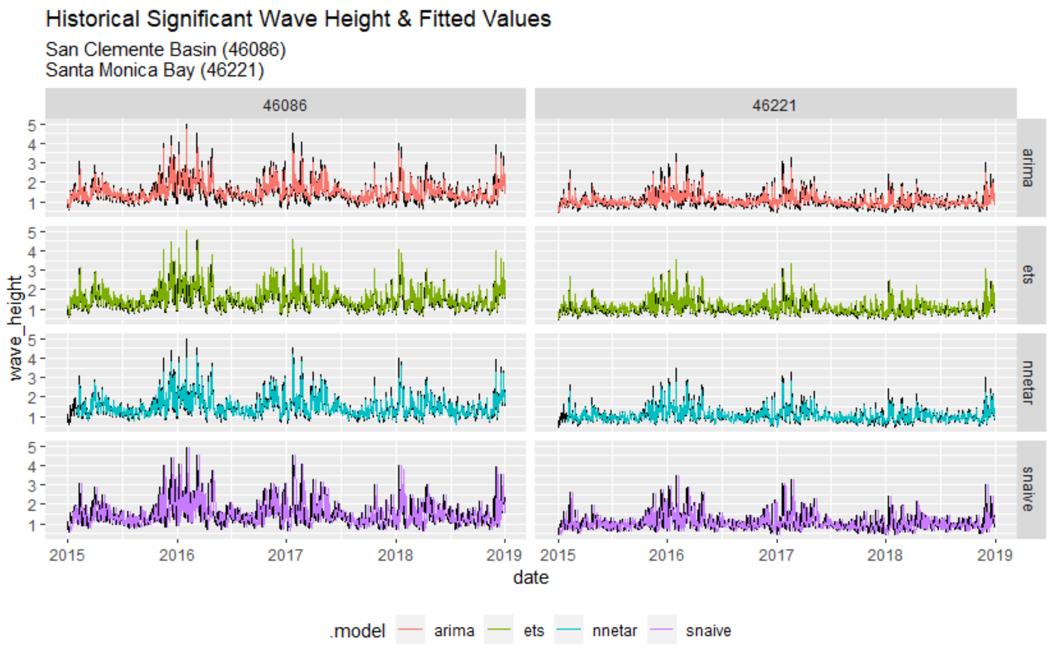

Figure 11: Time series model fits

These models were then evaluated against the ten day training data. The graphic below shows all five models plotted against the actual values.

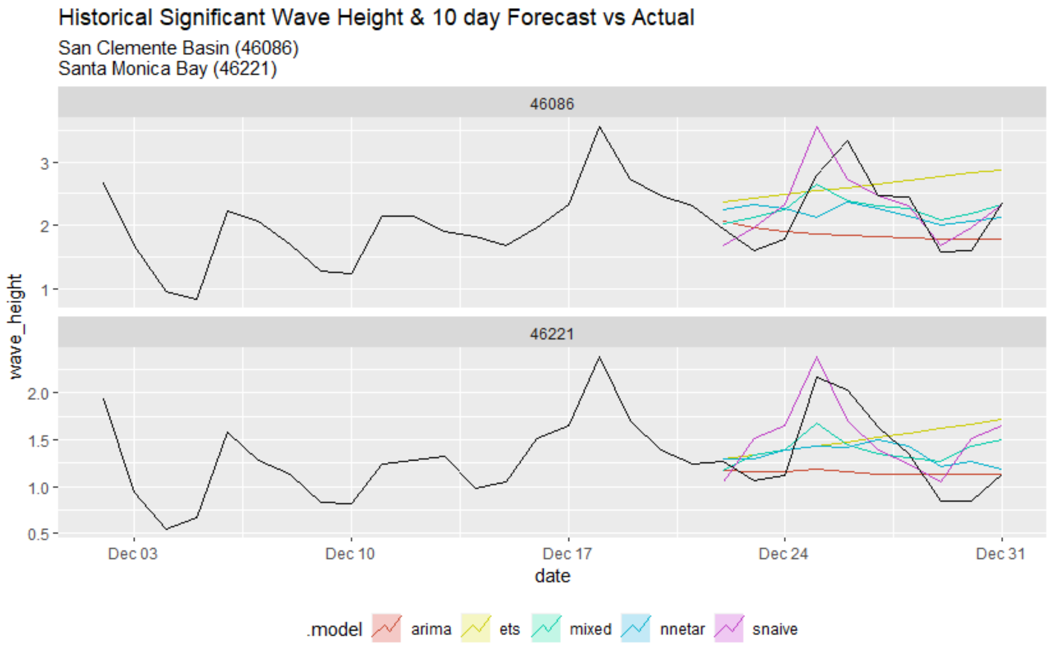

Figure 12: Time series model vs test data

Machine Learning Models

Machine learning for this case study was done using one tree based algorithm: XGBoost. The XGBoost algorithm was employed using the `xgboost` package. The data that was prepared for time series activities was portable to those for machine learning and the specialized time indexed data frame previously used was merely converted to a non time indexed data frame. From here two methods were explored, both using XGBoost, however taking into consideration two different types of feature engineering:

1. Temporal features e.g. day, month, year, quarter etc.
2. Lagged features of the original time series

For the first method a train test strategy was employed where rather than using $n - 10$ days of training data instead $n * .80$ of the data was used for training, with the balance of data used for testing. The training data was not chosen at random but instead chosen as a cutoff point in time that represented the first $n * .80$ of the data, which in this case was about at *2018-03-14*. Once the data for training was identified the training data frame of two variables, *time* and *wave height*, was passed into a function from the `timetk` package that automatically created time based features using the *time* variable. After time based features e.g. month label, weekday label, month, week, etc. were added the data frame that began as two variables increased to a data frame of 48 variables including the original *time* and *wave height* features. Features such as those denoting hour, minute, or seconds were removed as they were not useful, considering that day was the grain of time used in the case study.

The data frame with derived temporal features was then passed to the XGBoost algorithm. Setting up the algorithm was done using tools in the `parsnip` package. The XGBoost algorithm was set up to make regression based predictions since the outcome variable is the numeric representation of *wave height*. The algorithm boasts a wide variety of parameters and for this first method six were used. Below is a table of the six with a general description of its function:

Table 2: Temporal feature ML model parameters considered

parameter	description
mtry	How many columns to use to prevent over fitting
trees	The number of trees to grow
min_n	Minimum values per node
tree_depth	Maximum tree depth
learn_rate	Accuracy lever
loss_reduction	Model improvement required for a split

Arbitrary values were chosen for each of parameters in the table above, then the model was employed against the training data set. Once the model was trained it was assessed against the values of the test region representing the last $n - (n * .80)$ values. Using the San Clemente Basin buoy as an example, a visual representation of this schema with results can be seen below :

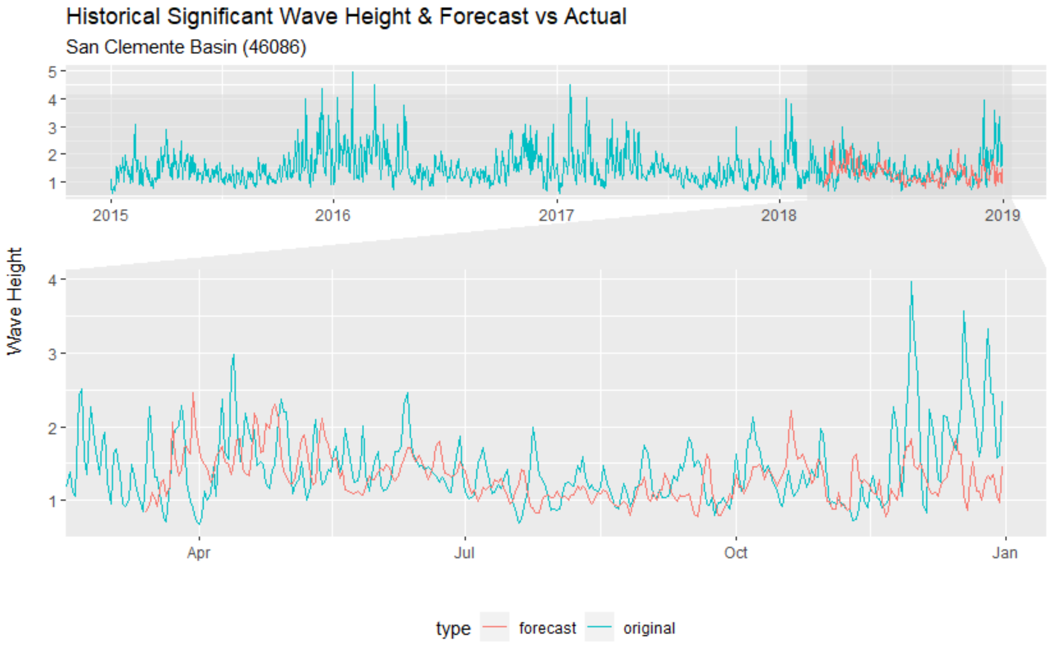

Figure 13: XGBoost temporal feature model train teset schema

For the second method a different strategy was employed which generally involved developing and working with lagged versions of the time series. The `forecastML` package was instrumental for implementing this method and provided the framework to carry out this methodology on the set of data gathered for this case study. The set of variables used from the buoy data collected for this method included: *date, wave_height, buoy_id, lat, lon, day, year, wind_spd*, and *sea_surface_temperature*. In addition two features were also added *day* and *year*. The training and testing strategy for this method began with creating a training data set using a function from the `forecastML` package which uses several parameters to determine the engineered features. For example, identifying the forecast horizons, in this case 1 and 10 days, the lagged features, here 1 to 30 days as well as annually, the frequency of the time series (days), features such as *day* and *year* which need to be accounted for but not lagged, what denotes a group (buoy id), and any features that do not change through time (latitude & longitude). Training data sets with newly engineered features were created for both forecast horizons i.e. 1 and 10 days.

Once the training data was set up and built, the next step was to develop a cross validation schema using the idea of windows in time. For this case study two windows were chosen and so the first window would be used to train and test and a validated on the "held out" window, and vice versa. With this method more specifically the first window would have a random set of data chosen as the training set and the test set being the balance of data within the same window. This was done for each forecast horizon that was identified i.e. one and ten days. The validation window then provided an honest assessment of model performance.

A visual representation of this can be seen below where the full year for 2015 and 2018 were used as windows in the previously described cross validation schema.

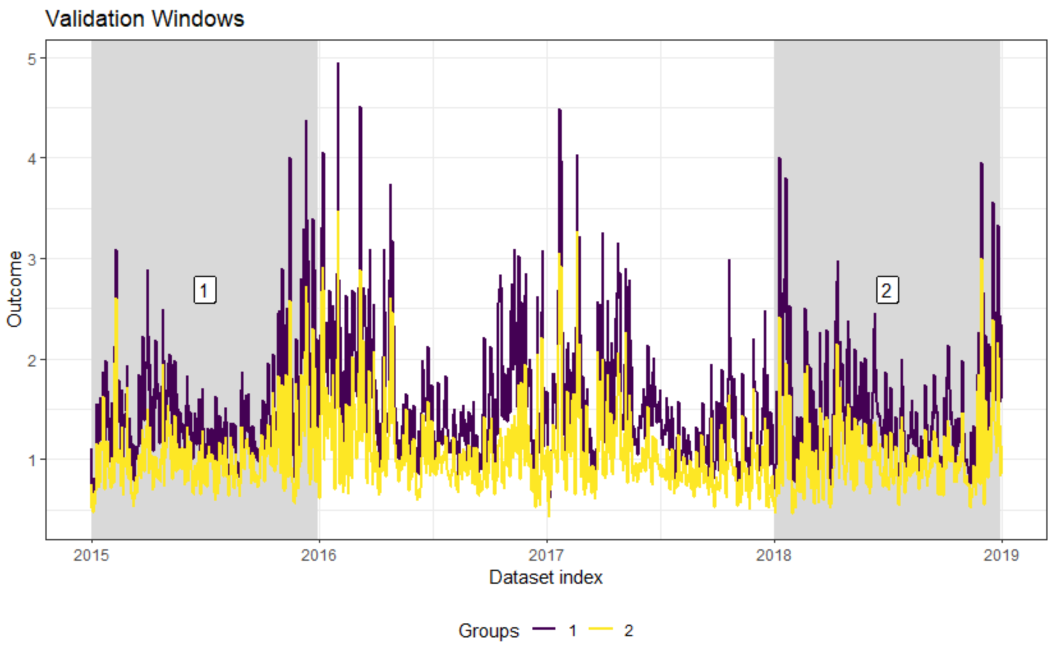

Figure 14: XGBoost lagged feature model train test schema

With the cross validation windows defined the framework of `forecastML` called for a user defined modeling function. The function was set up to use XGBoost as the "model engine", also within the function, model specific train and testing data sets were defined where a random sample of 80% of the data would be used to train and the balance for testing. Each of these data sets were set to be created within the modeling function with `xgboost` functions that create matrices of data specific to the XGboost algorithm. Parameters used with the XGBoost algorithm are specified in Table 3 along with a brief description of each.

Table 3: Lagged feature ML model parameters considered

parameter	description
max.depth	controls depth of tree
nthread	number of cores to use
nrounds	maximum number of iterations
metrics	metric to evaluate model accuracy
early_stopping_rounds	max number of iterations after no performance improvement
watchlist	Train and test data sets used to evaluate rounds

The parameters chosen allow for a method of evaluating the performance of the model during each round using a train and test method where each round is evaluated using the root mean squared error. In addition it also accounts for possible over fitting of the model by telling it to stop after a maximum number of rounds if there isn't any further improvement.

Once the user defined modeling function was defined it was passed as an argument to a `forecastML` function that uses this user defined modeling function on the previously created training data sets for both forecast horizons (1 and 10 days), and on each validation window that was defined. This created 4 models i.e. two forecast horizons multiplied by two validation sets used in evaluating the performance of this method of forecasting ocean waves. The models were all conveniently stored in one object. A user defined prediction function specific to the XGBoost algorithm was then built so that it could be passed to a more general predict function, along with the training data, which made predictions, using the object with four models, on the two validation sets for each forecast horizon.

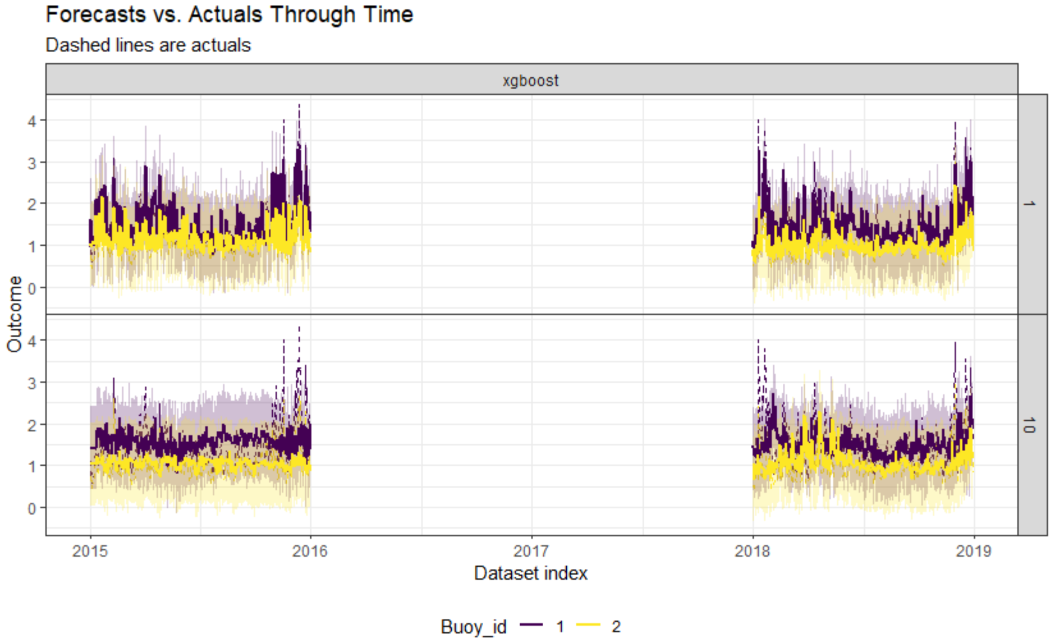

Figure 15: XGBoost lagged feature model validation forecast vs actual

In general the result of this process to evaluate performance can be seen in the visual above where the predictions for each validation window as well as for each forecast horizon

(1 and 10 days) are displayed against actual data.

Next a new forecast data set which using a function from `forecastML` to create a data set with features that are forward facing and creating one for each forecast horizon i.e. one and ten days. Forecasts were made for one and ten days into the future each using its respective xgboost model for the corresponding forecast horizon. Two forecast horizons for two validation sets produced two forecast for each buoy for each forecast horizon. This can more easily be seen in the graphic below:

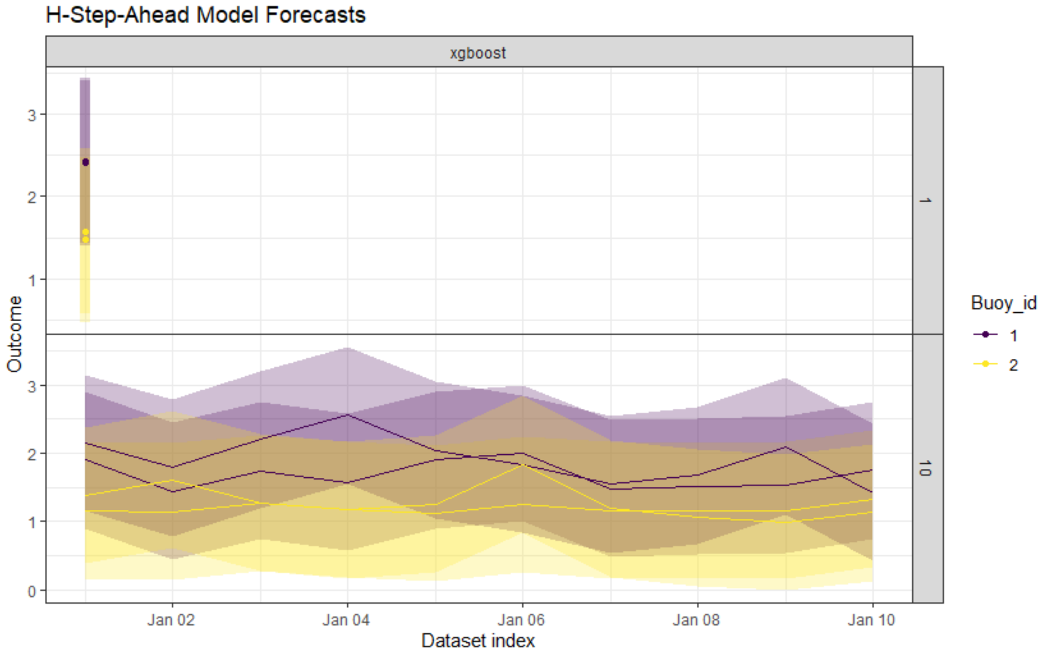

Figure 16: XGBoost lagged feature model CV step ahead forecasts

In the process of creating data for the xgboost forecast algorithm, the five digit buoy ids were converted to factors where buoy id one and two represent the San Clemente Basin and Santa Monica Bay buoys respectively.

In the last step of the `forecastML` framework the xgboost model was trained on all data. This meant that rather than train on validation windows, one window was created where the xgboost model would again be trained for each forecast horizon for each buoy. This new window that covers the entire time series can be seen below:

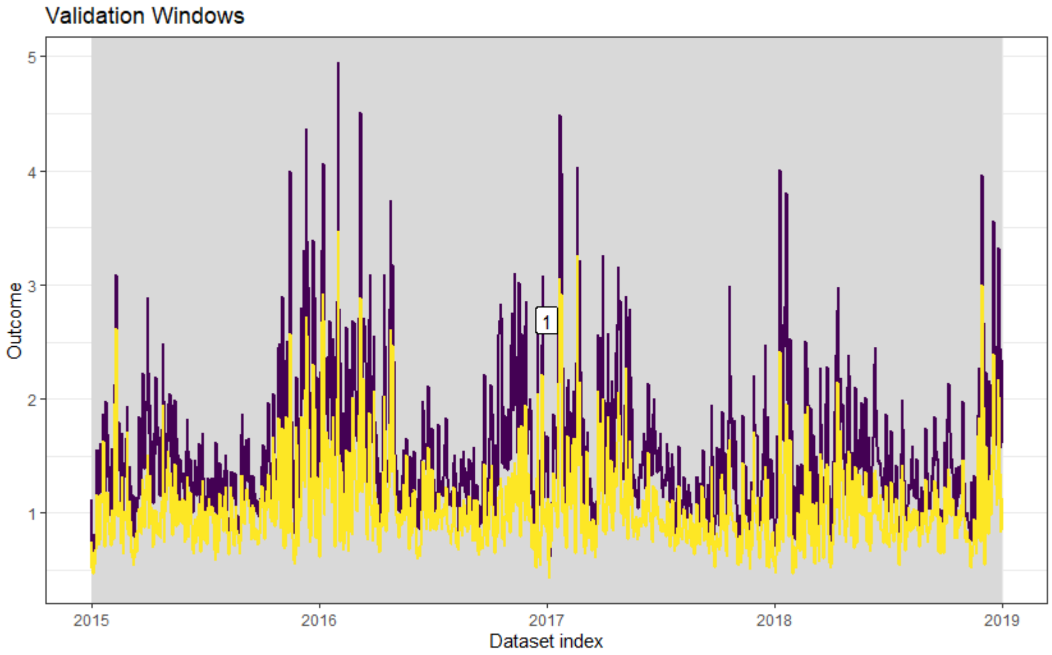

Figure 17: XGBoost lagged feature model train on all data window

The models created in this step were then combined for each forecast horizon. This meant that for the ten day forecast, the forecast was a composite of each step ahead horizon forecast, with the one day model forecasting one day ahead and the ten day model ten days ahead, and both combined for a ten day forecast.

Empirical Dynamic Modeling

During the course of methods research for this case study a different framework that neither fell into the classical time series or machine learning camps was briefly explored: *Empirical Dynamic Modeling* (EDM). Chang et al. (2017) describes EDM as as a framework that takes into account the complexities of natural systems, the possibility of spurious correlations, and the changing nature of variable interaction over time with changes in system state. EDM can ultimately aid in forecasting activities of said systems. Furthermore Ye et al. (2019) explain that contrary to parametric equations EDM instead aims deduce from the data relationships and patterns where the time series can be though of casting a shadow of sorts and from this EDM works to model the complex system.

In their article Ye et al. (2019) provide a visual representation (see below) of this general idea of a complex system projecting a pattern over time.

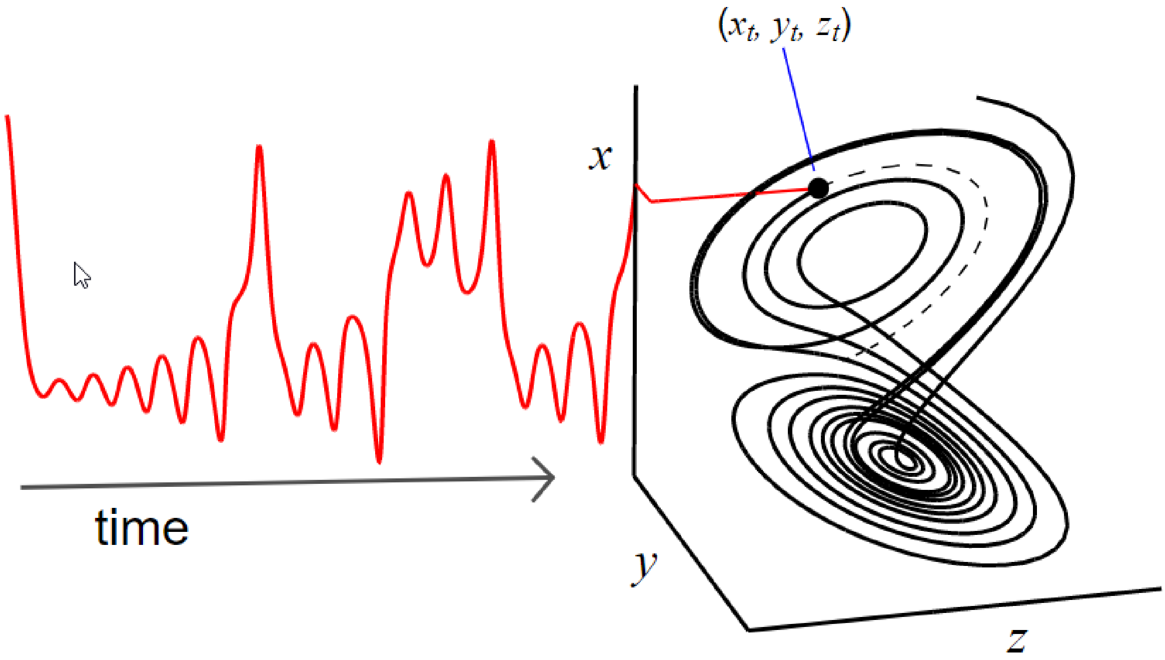

Figure 18: Time Series Projection from the Lorenz Attractor by Ye et al. (2019)

$$\mathbf{x}_t = (x_t, x_{t-\tau}, \ldots, x_{t-(E-1)\tau})$$

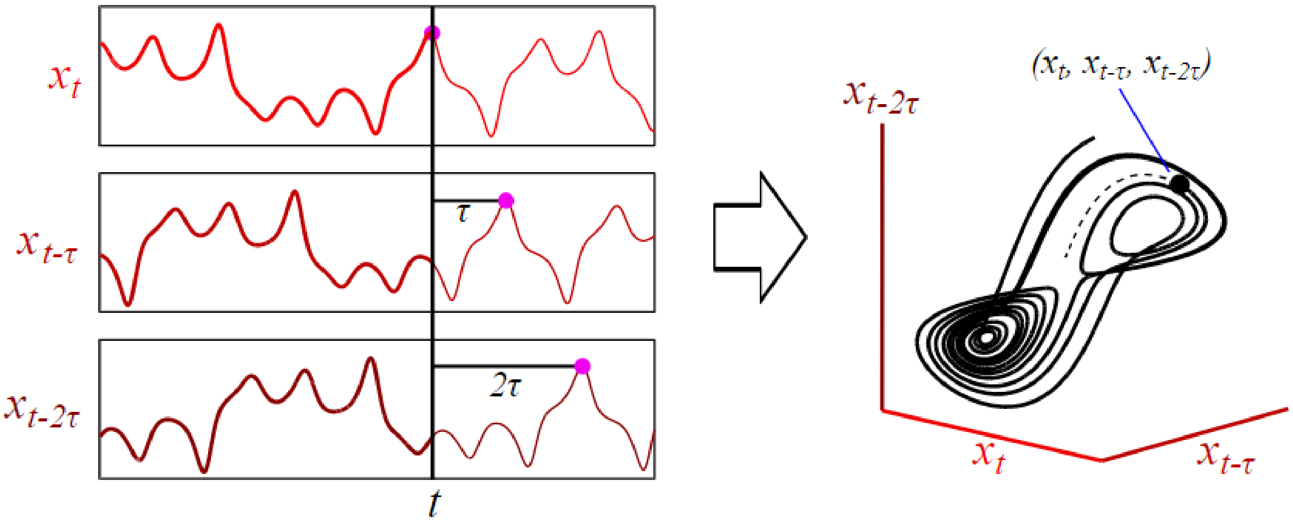

Figure 19: Attractor Reconstruction from 3 Lagged Coordinates by Ye et al. (2019)

In addition Ye et al. (2019) go on to explain that not only can a single projection be taken from this type of dynamic system but also those of various lags (E) which are used in the EDM framework to infer characteristics of the system as seen in Figure 19 above.

For this method the `rEDM` package was used, specifically employing an approach presented by Sugihara and May (1990) called Simplex Projection to both produce forecasts as well as examine performance. The base data that was used for the time series and machine learning methods was reused here as well, however this time a single vector, or sequence, of values was the only input needed. Data for the San Clemente Basin was used as an example. A train/test strategy was also used, where the first $n - (n * .20)$ values of the data were used as training data and with the last $n * .20$ used as testing. The partitioned training data set along with the original full vector of values of *wave height* were passed to the simplex function. The function was also given a set of "embedding values" or values of E between 1 and 10 for the function to evaluate each value of E using leave one out cross validation on the entire time series. "Forecast skill" as the measure of correlation (rho) between the predicted and observed values was then be compared to each value of E to determine the best one.

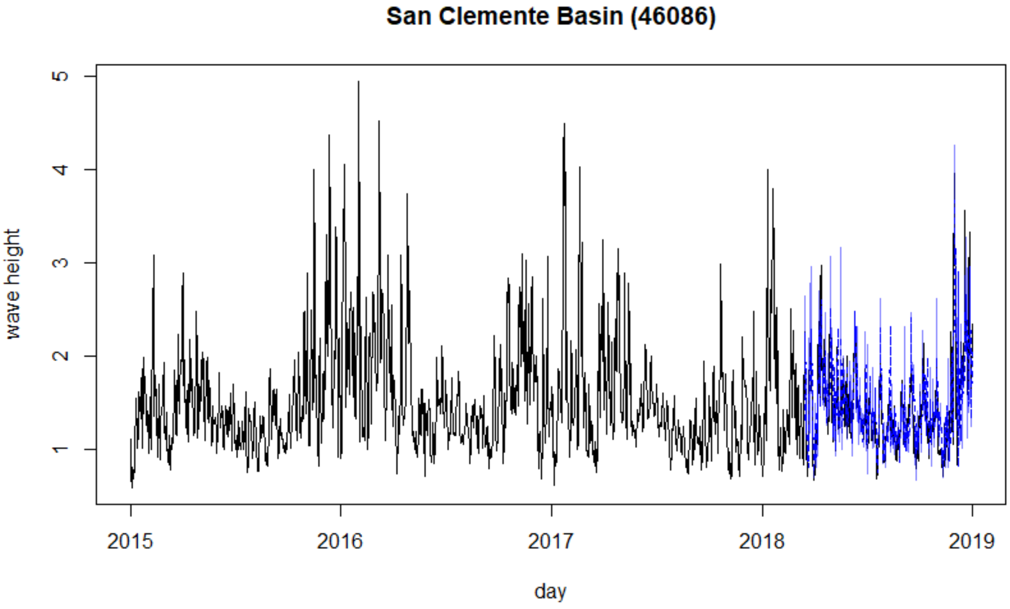

Figure 20: EDM Forecast vs Actuals

The output of this function suggested that the optimal value of E is 3, corresponding to the highest value of *rho*. This new value of $E = 3$ was then passed into the simplex function again along with the test data set so that predictions vs actual values could be compared. Figure 20 above shows the forecast values vs actual on the last 20% of the data.

Methodology Rational

For this case study an in depth explanation of the mathematics behind each model is out of scope as the core underlying methods are all well studied and documented. Complex formulas will be less of a concern and more so practical application, specifically how methods of time series, machine learning, and to a some extent empirical dynamic modeling can be used to model and forecast ocean buoy data using data from the San Clemente Basin and Santa Monica Bay buoys. However, a general overview of each of the methods is prudent to help increase understanding of their application. I'll begin with time series models, move to the machine learning frame work used, and finally discuss the implications of empirical dynamic modeling on this case study.

With the classical time series methodology, four methods were used: exponential smoothing, ARIMA, seasonal naive, neural network, and an ensemble of the first three not including the neural net. Generally speaking exponential smoothing models are a method of forecasting where weights are given to past observations which help determine their effect on future observations, with more recent observations lending more weight and past observations exponentially less (Hyndman and Athanasopoulos, 2018). Furthermore Hyndman et al. (2008) puts together a matrix of models that also take into account trend, seasonal and error patterns, for a total of 18 possible state space models (three trend possibilities multiplied by three seasonal provide 9 methods multiplied by two possible error patterns). It was shown in Hyndman et al. (2002) that ETS performs well for short term forecasts with seasonal variation.

ARIMA modeling takes into account three items: autogression, or the relationship a time series has with lagged versions of itself, stationarity, or the degree to which a time series is independent of its time index, and finally a moving average component, which refers to an average of past errors. What Hyndman and Athanasopoulos (2018) shows is that ARIMA models are also capable of handling seasonal characteristics of a time series and notes that both ETS and ARIMA models are both complementary and frequently used methods of forecasting a time series. These methods therefore provide a solid foundation for this case study when analyzing a univariate time series of *significant wave height*.

The seasonal naive model is categorized in Hyndman and Athanasopoulos (2018) as a simple forecasting method that uses as its forecast the value of the previous season. For this case study this the seasonal period was automatically set to seven so that a future value would be represented by what it was seven days ago. The simplicity of of this type model is a main benefit as it can serve as a baseline for other methods as well as to help account for and express variation in an ensemble model that may be smoothed over in other models.

With a neural network autoregression the model is developed with lagged values of itself and uses a single hidden layer which takes as inputs the lagged values and where a series of computations are performed including weighting and transforming input values before they are passed to the output layer. These hidden layers Hyndman and Athanasopoulos (2018) notes are what enable the model to account for non-linear patterns in the data and also help to weaken the effect extreme values may have. Extensive work has been done using neural networks to predict *significant wave height*, for example in Londhe and Panchang (2005) one day forecasts at five locations using neural networks along with ocean buoy data were shown to be fairly accurate for shorter lead times as well as providing rapid computational implementation. Additionally Vimala et al. (2014) showed that using buoy data forecast with neural networks was successful for short lead times and given the results neural nets were considered for real time forecasting activities.

The fifth model used in the classical time series frame work was an ensemble of the exponential smoothing, ARIMA, and seasonal naive forecast models. The ensemble in this instance is denoted as a simple average of the aforementioned models. Research for ensemble models to predict ocean waves is ongoing, for example both Chen (2006) and Cao et al. (2007), note that in general ensemble forecasts tend to produce favorable results and with Kumar et al. (2018) more recently showing ensemble modeling outperforms methods that don't use this technique which in general works to combine multiple forecasts. Given these advancements the ensemble approach was also given consideration in this case study.

A core activity of time series analysis is making sure that methods are tested for accuracy in an honest way, meaning that models trained on data are tested on subsequent data that has been held out from modeling activities. Hyndman and Athanasopoulos (2018) suggests that 20% of the data is frequently held out as a test data set, but that this may vary, and ultimately the amount of data used for testing at minimum should be a long as the planned step ahead forecast. In this case the step ahead forecast was 10 days, meaning models would be built to forecast 10 days into the future. Taking into account the forecast horizon the test data set was then made the same length, or ten days. In James et al. (2013) it's noted that models tend to perform poorly as the amount of data they are trained on drops, and so using a good majority of data for training the time series models was warranted. In terms of accuracy in Hyndman et al. (2006) a parsimonious method that works well for evaluating one series is the mean absolute error (MAE). This method is also noted by Ben Freestone (2019) as a great starting point and appropriate for surf forecasts because it is easily understood by consumers of the forecast who depend on knowing scale dependent errors. MAE was therefore used in this case study to assess forecast accuracy for *significant wave height*.

The `fpp3` package and the additional packages it loads provide an extensive suite of time series forecasting algorithms and excellent utilities for structuring time series data, including those that include various grouping levels. What this means is that two for more univariate time series can be represented as the ubiquitous data frame object, as well as enhanced versions of the data frame, that allow of easy manipulation of time based data e.g. row based aggregation, filtering, grouping, feature engineering, and single object input to modeling functions with multi output forecasts. This is precisely the structure that was used for this case study with buoy ids denoting distinct groups within the data frame structure. This method of dealing with time based data is beneficial because it can easily fit into the data science framework proposed by Wickham and Grolemund (2016), where rows are observations and columns variables, making this data portable to other methods that are not exclusively time based such as those in machine learning as well as various visual methods, all within the open source *R* framework.

Two machine learning methods were employed in this case study. The first used temporal features e.g. day, week, month, while the other used features that in general were lagged values of the outcome variable *significant wave height*. For both methods the time series forecast was framed as a supervised learning problem i.e. machine learning, where each value of the response variable has one or more associated predictor variables, and the overall goal is to model the relationship between both in order to understand these relationships and ultimately produce predictions (James et al., 2013). The machine learning engine used for this case study was XGBoost which is excellent for a wide array of problems, scalable, fast, and efficient in terms of computational resource usage (Chen and Guestrin, 2016). Since there are many buoys, each collecting vast amounts of data, XGBoost, given its aforementioned strengths was a great candidate for this case study.

In the first machine learning approach a schema of train and testing was followed with the first 80% of data used as training and the last 20% used as testing which is in line with the split suggested by Hyndman and Athanasopoulos (2018) as being typical for a time series. In addition Silva (2014) showed that a feature engineering approach based on temporal characteristics performed well in forecasting a natural process time series and therefore in this first approach several features were created that were derived from the *time* variable itself. Because the predictor variables are time stamps in and of themselves, forecasting was a relatively simple procedure wherein future time stamps were created along with their associated derived temporal features e.g. day, month, year, quarter etc. and these predictor variables were then used along with a single XGBoost model to make forecasts.

In contrast to the first machine learning approach, the second was implemented given research material put forth in Bergmeir et al. (2018), and implemented in the `forecastML` package, which explores strategies for employing cross validation techniques that are not time specific coupled with autoregressive features of a time series. The forecast strategy for this second method is a variant of what current literature calls "multiple output" forecasting, where multiple models are used to forecast specific horizons. Literature has shown that this method of forecasting can be robust against the bias inherent iterative models such as those found in classical time series(Marcellino et al., 2006). Additionally Taieb et al. (2012) show via a thorough review of various forecast methods that multiple output forecasting frameworks perform well and so the implementation put forth in `forecastML` as used in this case study.

There are several publications that explore non linear dynamics and its applications to natural systems, for example Runge et al. (2019), Sugihara et al. (1990), and Sugihara (1995) provide an extensive foundation to support this theoretical framework. In addition there is excellent work being done at the Sugihara Lab at the Scripps Institution of Oceanography, much of which is based on the premise that natural systems cannot simply be modeled through application of equations per se, rather complex non linear systems require capturing the patterns from the natural system itself and using this as a foundation understanding and predicting events in natural systems such as a river, lake, or perhaps in this case the ocean and ocean waves as represented by data captured *in situ* by ocean buoys. The `rEDM` package is a well documented tool in the *R* ecosystem that was designed to implement the ideas of Empirical Dynamic Analysis. In addition, the cost of code implementation i.e. effort to code and produce results, was low when compared to the methods used in the previously discussed time series and machine learning frameworks.

Deep learning frameworks were considered for this case study as they are, in the field of Data Science, the bleeding edge of machine learning algorithms. Deep learning requires a new paradigm of thinking about data structure in a 3D format that is not as ubiquitous or easily understood as that of the data frame. For example in Dancho and Keydana (2018) the entire workflow to predict sunspots is several orders of magnitude higher in code cost (the time and effort to implement and retrieve results) than the same analysis on the same data set but using the `rEDM` framework, which produces comparable, if not sightly better (visually speaking) results from a train test schema. Therefore, in an effort to provide a Data Scientist with methods that are low in “code cost”, this method was not used and instead comparatively more parsimonious methods such as those discussed above were used for this case study.

Findings/Results

The goal of this case study was to produce step ahead forecasts of *significant wave height* using ocean buoy data. Two main methods were used to achieve this aim: classical time series, and supervised learning/machine learning. In addition, to a lesser extent, empirical dynamic modeling was also explored. A forecast will more than likely never be entirely correct and as such a good assessment of a forecast activity is to measure error. Error for these methods was measured on test and validation sets depending on the method being used. The error was calculated as $e_t = Y_t - F_t$, more specifically using the Mean Absolute Error (MAE) defined simply as $mean(|e_t|)$.

For the five time series models used, the training set represented the first $n - 10$ days of data and the test the last $n - (n - 10)$ with n being the total number of days. MAE was then calculated for each buoy model combination. The results of this measure of error can be seen in Table 4 which shows, for example in the first two rows, the two entries for the forecast method, ARIMA. Each method then shows its corresponding buoy id for which the forecast was made, and finally in the third column the forecast error measure (MAE). Since there are five models that were used and two buoys, there are a total of ten rows reporting

MAE. The lowest MAE belongs to the neural network time series model for buoy id 46221 and the highest to the ETS model for buoy id 46086. In terms of methods the mixed model and seasonal naive models performed the best overall for both buoys.

Table 4: Time series training set MAE

model	buoy_id	MAE
arima	46086	0.527
arima	46221	0.337
ets	46086	0.635
ets	46221	0.435
mixed	46086	0.358
mixed	46221	0.342
nnetar	46086	0.477
nnetar	46221	0.293
snaive	46086	0.319
snaive	46221	0.348

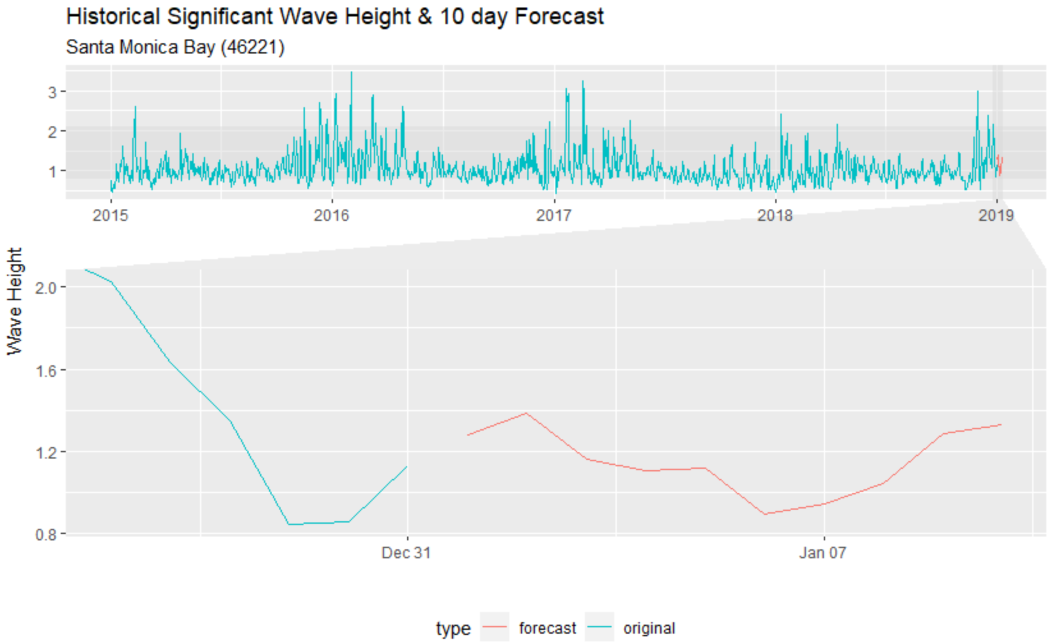

Figure 21: Buoy id 46221 Ten Day Forecast time series nueral net

Figure 21 provides a visual example of the ten day forecast for the Santa Monica Basin buoy using the time series neural network model.

The next model that was examined on the basis of training error was the XGBoost model that used temporal derived features e.g. month, day, week, quarter etc. This model was trained on the first $n * .80$ of the data and the last $n - (n * .80)$ used as test data. Table 5 shows the error for this method for both buoys. The XGBoost model for buoy id 46221 has a lower MAE than the model for buoy id 46086.

Table 5: XGBoost temporal feature training set MAE

model	buoy_id	MAE
xgboost	46086	0.394
xgboost	46221	0.226

An example of a ten day forecast for the XGBoost temporal features model is shown below.

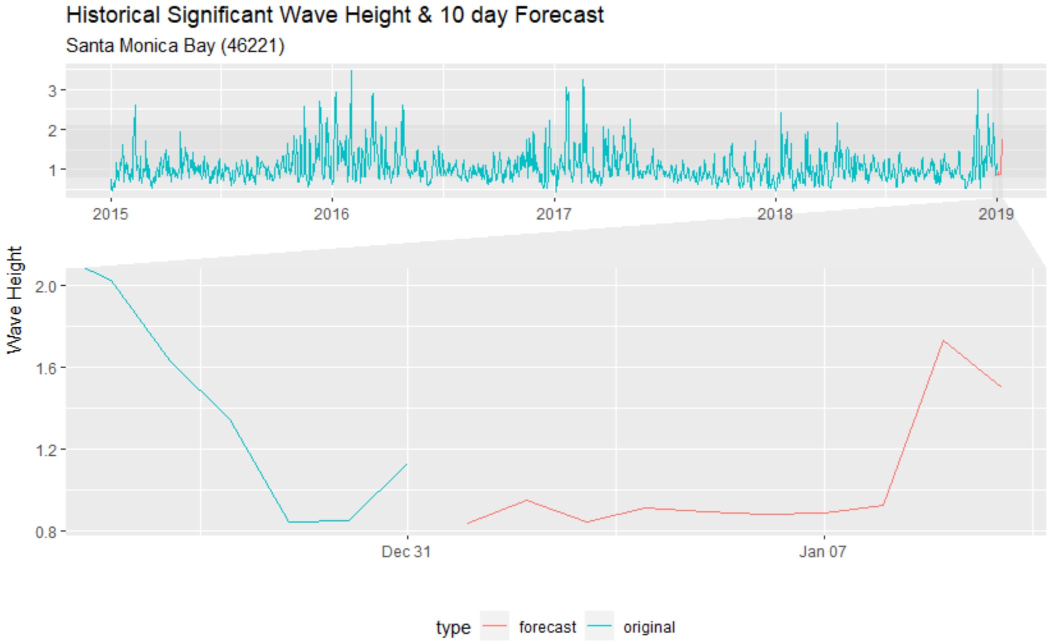

Figure 22: Buoy id 46221 Ten Day Forecast xgboost temporal features

The final method that was examined on the basis of error was the xgboost model that used lagged features as engineered predictor variables. This method employed a nested CV method which produced forecast errors for various facets of the data. For example at the highest level forecast error was examined on the basis of buoy id alone. The next level was forecast error for buoy id and forecast horizon, and lastly, error reported by buoy id, horizon, and validation window.

Table 6: XGBoost lagged feature global error

model	buoy_id	MAE
xgboost	46086	0.374
xgboost	46221	0.259

Table 7: XGBoost lagged feature horizon error

model	buoy_id	forecast_horizon	MAE
xgboost	46086	1	0.293
xgboost	46221	1	0.228
xgboost	46086	10	0.446
xgboost	46221	10	0.280

Table 8: XGBoost lagged feature window error

model	buoy_id	forecast_horizon	window_number	MAE
xgboost	46086	1	1	0.317
xgboost	46221	1	1	0.263
xgboost	46086	1	2	0.269
xgboost	46221	1	2	0.192
xgboost	46086	10	1	0.460
xgboost	46221	10	1	0.255
xgboost	46086	10	2	0.431
xgboost	46221	10	2	0.305

Tables 6,7, and 8 all show the measures of error (MAE) for the various facets of time, validation window, and forecast horizon produced by the `forecastML` framework coupled with an XGBoost model. Figure 23 shows the results: a 10 day step ahead forecast using the XGBoost lagged feature modeling approach.

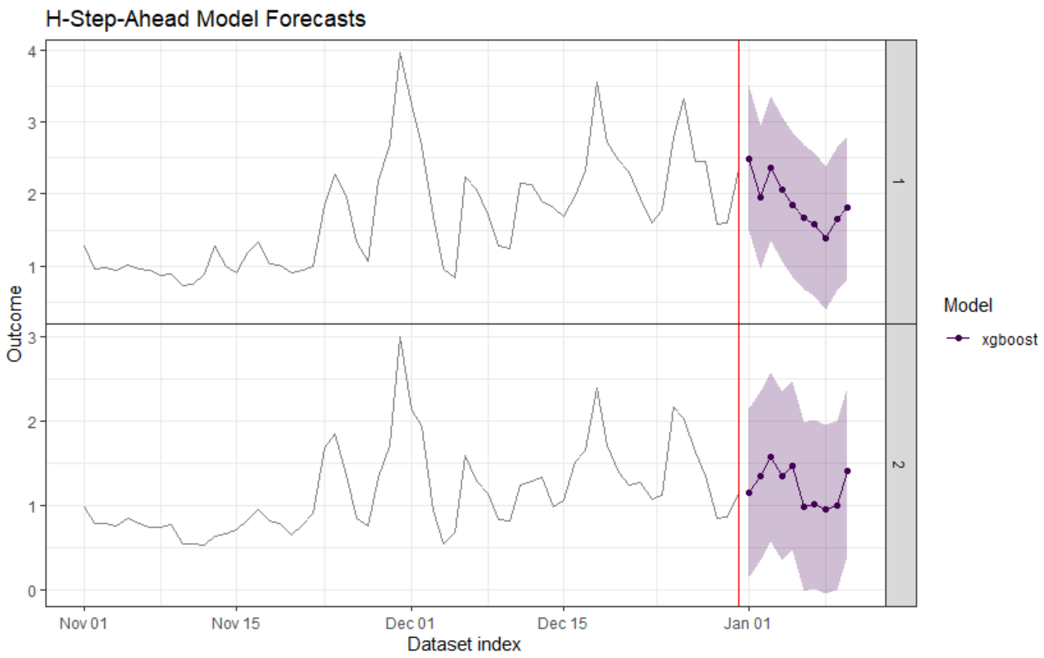

Figure 23: Ten day forecast xgboost lagged feature model

The last method that was explored (to a lesser extent) did not fall into the classical time series or machine learning frameworks but rather used the theoretical method of Empirical Dynamic Modeling. Error terms (MAE) were calculated by using the first $n * .80$ of data to train while the last $n - (n * .80)$ was used for testing. MAE was calculated for each buoy from simplex projection forecasts on the training data (Sugihara and May, 1990). Table 9 summarizes the results below and shows buoy id 46221 with the lowest MAE between the two buoys.

Table 9: Empirical Dynamic Model Simplex Projection error

model	buoy_id	MAE
simplex	46086	0.272
simplex	46221	0.184

Ten day forecasts were produced by the time series and machine learning models as well as helpful error measures to assess forecast accuracy. Error for the time series models was most stable between buoys using the mixed or ensemble model. For the temporal feature machine learning model, error was lower for the Santa Monica bay buoy compared to the San Clemente Basin. The lagged feature machine learning model was similar with lower error for the Santa Monica Bay buoy. In addition, because the lagged feature model was built by combining models that produced one and ten day forecasts respectively, error was also reported by these two facets of time and on two validation sets. Error was generally stable across both validation windows for both buoys, with the Santa Monica Bay error being lower for both. Forecast error was also lower for Santa Monica Bay when compared on the basis of both one and ten day forecast horizons, and was also ultimately lower in terms of global error. Training error from empirical dynamic modeling was lower for the Santa Monica Bay buoy compared to San Clemente Basin. Empirical dynamic modeling exhibited the lowest error for both buoys when compared to the measures of error from the time series, temporal feature machine learning model, and the global error for the lagged feature machine learning model.

Recommendations/Next Steps/Conclusion

For this case study the objective was to produce ocean wave forecasts using *significant wave height* data collected by ocean buoys. Rather than use physics based models which require a significant amount of oceanography and ocean engineering domain expertise to employ and use, two techniques, well within the domain of data science, were instead used: classical time series and machine learning. In addition, the theoretical frame work of empirical dynamic modeling, which focuses on modeling complex systems from patterns inferred from observed data, was examined but to a lesser extent. The objective of analyzing and forecasting *significant wave height* using the *R* ecosystem, ocean buoy data, time series, machine learning, and empirical dynamic modeling was successful. Both time time series and machine learning techniques produced reasonable ten day step ahead forecasts that were within the bounds of observed *significant wave height* values.

The results from the times series modeling suggest that using an ensemble approach provides a balanced forecast of *significant wave height* for both the San Clemente Basin and Santa Monica Bay buoys. A machine learning approach using temporal features engineered from the date index of a *significant wave height* time series produced comparable results, however took much less processing time to compute using the XGBoost algorithm. Using XGBoost along with lagged features of the outcome variable, along with a direct forecast approach, proved not only fast but also promising for further study. For example, a schema where multiple algorithms (beyond perhaps just XGBoost) are used to forecast specific horizons of an h step ahead forecast, where each contribute their best performance to the appropriate horizon. However, using XGBoost alone in this framework still produced reasonable forecasts. Although empirical dynamic modeling was not used to make step ahead forecasts it did perform very well in terms of error and was lower overall for each buoy compared to all time series and machine learning techniques.

This case study aims to provide a data scientist with a foundation for retrieving ocean buoy data, studying, modeling, and forecasting ocean waves using concepts in classical time series, machine learning, as well as providing and introduction into the application of empirical modeling. Classical time series techniques provide a multitude of tools to examine, both visually and statistically, characteristics of ocean waves that can help a data scientist increase their understanding of ocean wave dynamics. XGBoost machine learning techniques are both scalable and computationally efficient and can be used to train models using engineered features from the times series itself, physics model output, or perhaps even human observations. And given the flexibility of empirical dynamic modeling and its ability to model complex non linear systems such as the ocean and ocean waves EDM has proven to be an excellent method for a data scientist to use when modeling and forecasting ocean waves. All of these methods have excellent implementations within the R statistical programming ecosystem and are therefore well within the preview of a data scientist interested in ocean wave forecasting.

This case study used standard meteorological data measured by two ocean buoys in the Southern California region and therefore modeling is limited to observations used from the Santa Monica Bay and San Clemente Basin buoys as daily averages of hourly data. Different regions may exhibit different characteristics which may effect the model parameters used which could be region specific. In addition the availability and reliability of buoys in various regions constrain study efforts based primarily on buoy data. Methodological choices were made according to the availability of a programmatic implementation in the *R* statistical programming ecosystem and not other platforms such as *Python*. It is beyond the scope of this case study to provide an accounting of the mathematical framework of a model and instead focuses on practical application. The techniques used are well studied and documented to aid further research in different regions, using different time grains (e.g. hour, month) or perhaps studies into the intricacies of the algorithms themselves e.g. hyper parameter tuning or how emerging techniques such as empirical dynamic modeling can improve ocean wave forecasting.

Ocean buoys and the meteorological data they collect are frequently used as the "ground truth" that support activities which require accurate and at times live measurements of ocean waves. In addition, the data ocean buoys collect is used to help verify various new and cutting edge ocean wave forecasting techniques that benefit public safety and both private and government interests. With this in mind, ocean buoy data in and of itself was coupled with the *R* statistical programming ecosystem to support and develop this case study on ocean wave forecasting using classical time series and machine learning methods, in addition to briefly exploring new theoretical methods. The methodologies used and workflows developed produced reasonable forecasts when measured on out of sample observations and may ultimately prove to be a viable framework well within the domain of data scientists who wish to study and forecast ocean waves.

References

Behrens, J., Terrill, E., and Thomas, J. (2018). Cdip wave observations during hurricanes irma, jose, and maria, and a noreaster. *Shore and Beach*, 86(3):14–20.

Ben Freestone (2019). *Surf Forecast Accuracy.* https://medium.com/surfline-labs/surf-forecast-accuracy-b563605f104c [Accessed: July 7].

Bergmeir, C., Hyndman, R. J., and Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83.

Bidlot, J.-R., Holmes, D. J., Wittmann, P. A., Lalbeharry, R., and Chen, H. S. (2002). Intercomparison of the performance of operational ocean wave forecasting systems with buoy data. *Weather and forecasting*, 17(2):287–310.

Cao, D., Chen, H., and Tolman, H. (2007). Verification of ocean wave ensemble forecast at ncep. In *Proc. 10th International Workshop on Wave Hindcasting and Forecasting and Coastal Hazards Symposium.*

CDIP, Scripps Institution of Oceanography (2019). *CDIP Documentation.* https://cdip.ucsd.edu/m/documents/index.html [Accessed: May 26].

Chang, C.-W., Ushio, M., and Hsieh, C.-h. (2017). Empirical dynamic modeling for beginners. *Ecological Research*, 32(6):785–796.

Chen, H. (2006). Ensemble prediction of ocean waves at ncep. In *Proc. 28th Ocean Engineering Conference.* Citeseer.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Collins, S. (2020). *So how do we get surf?* http://www.surfline.com/surfline/lolaarchive/seans_surfology.cfm [Accessed: May 26].

Crosby, S. C., Kumar, N., O'Reilly, W., and Guza, R. (2019). Regional swell transformation by backward ray tracing and swan. *Journal of Atmospheric and Oceanic Technology*, 36(2):217–229.

Dancho, M. and Keydana, S. (2018). Rstudio ai blog: Predicting sunspot frequency with keras. https://blogs.rstudio.com/tensorflow/posts/2018-06-25-sunspots-lstm/ [Accessed: July 7].

Holthuijsen, L. H. (2010). *Waves in oceanic and coastal waters*. Cambridge university press.

Hyndman, R., Koehler, A. B., Ord, J. K., and Snyder, R. D. (2008). *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J. et al. (2006). Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4(4):43–46.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., and Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of forecasting*, 18(3):439–454.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

James, S. C., Zhang, Y., and O'Donncha, F. (2018). A machine learning framework to forecast wave conditions. *Coastal Engineering*, 137:1–10.

Janssen, P. A. (2008). Progress in ocean wave forecasting. *Journal of Computational Physics*, 227(7):3572–3594.

Janssen, P. A., Hansen, B., and Bidlot, J.-R. (1997). Verification of the ecmwf wave forecasting system against buoy and altimeter data. *Weather and forecasting*, 12(4):763–784.

Kumar, N. K., Savitha, R., and Al Mamun, A. (2018). Ocean wave height prediction using ensemble of extreme learning machine. *Neurocomputing*, 277:12–20.

Londhe, S. and Panchang, V. (2005). One-day wave forecasts using buoy data and artificial neural networks. In *Proceedings of OCEANS 2005 MTS/IEEE*, pages 2119–2123. IEEE.

Lovejoy, D. W. (2012). Ocean waves. *Earth Science: Earth's Weather, Water and Atmosphere*, 2:391–396.

Ludka, B., Guza, R., O'Reilly, W., Merrifield, M., Flick, R., Bak, A., Hesser, T., Bucciarelli, R., Olfe, C., Woodward, B., et al. (2019). Sixteen years of bathymetry and waves at san diego beaches. *Scientific data*, 6(1):1–13.

Mackenzie, B., Celliers, L., de Freitas Assad, L. P., Heymans, J. J., Rome, N., Thomas, J. O., Anderson, C., Behrens, J., Calverley, M., Desai, K., et al. (2019). The role of stakeholders and actors in creating societal value from coastal and ocean observations. *Frontiers in Marine Science*, 6:137.

Mahjoobi, J. and Etemad-Shahidi, A. (2008). An alternative approach for the prediction of significant wave heights based on classification and regression trees. *Applied Ocean Research*, 30(3):172–177.

Mandal, S. and Prabaharan, N. (2006). Ocean wave forecasting using recurrent neural networks. *Ocean engineering*, 33(10):1401–1410.

Marcellino, M., Stock, J. H., and Watson, M. W. (2006). A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *Journal of econometrics*, 135(1-2):499–526.

National Data Buoy Center (2018a). *Handbook of Automated Data Quality Control Checks and Procedures.* https://www.ndbc.noaa.gov/NDBCHandbookofAutomatedDataQualityControl2009.pdf [Accessed: June 7].

National Data Buoy Center (2018b). *Measurement Descriptions and Units.* https://www.ndbc.noaa.gov/measdes.shtml [Accessed: May 26].

National Data Buoy Center (2018c). *Programmatic Environmental Assessment for the National Oceanic and Atmospheric Administration National Data Buoy Center.* https://www.ndbc.noaa.gov/pea/ndbc_final_pea_20180104.pdf [Accessed: May 26].

O'Reilly, W. C., Olfe, C. B., Thomas, J., Seymour, R., and Guza, R. (2016). The california coastal wave monitoring and prediction system. *Coastal Engineering*, 116:118–132.

Rohweder, J., Rogala, J., Johnson, B., Anderson, D., Clark, S., Chamberlin, F., Potter, D., and Runyon, K. (2012). Application of wind fetch and wave models for habitat rehabilitation and enhancement projects–2012 update. *US Army Corps of Engineers, Contract report.*

Runge, J., Bathiany, S., Bollt, E., Camps-Valls, G., Coumou, D., Deyle, E., Glymour, C., Kretschmer, M., Mahecha, M. D., Muñoz-Marí, J., et al. (2019). Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):1–13.

Silva, L. (2014). A feature engineering approach to wind power forecasting: Gefcom 2012. *International Journal of Forecasting*, 30(2):395–401.

Sugihara, G. (1995). Prediction as a criterion for classifying natural time series. In *Chaos and Forecasting: Proceedings of the Royal Society Discussion Meeting. World Scientific, Singapore*, pages 269–294.

Sugihara, G., Grenfell, B. T., and May, R. M. (1990). Distinguishing error from chaos in ecological time series. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 330(1257):235–251.

Sugihara, G. and May, R. M. (1990). Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature*, 344(6268):734–741.

Taieb, S. B., Bontempi, G., Atiya, A. F., and Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, 39(8):7067–7083.

Thomas, J., McWhorter, J., and Olfe, C. (2016). Importance of wave observations for model validation within the san pedro bight, california. *Marine Technology Society Journal*, 50(3):69–71.

Venkatesan, R., Vedachalam, N., Joseph, K. J., and Vengatesan, G. (2019). Data returns and reliability metrics from the indian deep ocean wave measurement buoys. *Marine Technology Society Journal*, 53(6):6–20.

Vimala, J., Latha, G., and Venkatesan, R. (2014). Real time wave forecasting using artificial neural network with varying input parameter.

Wickham, H. and Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data.* " O'Reilly Media, Inc.".

Ye, H., Clark, A., Deyle, E., and Sugihara, G. (2019). redm: an r package for empirical dynamic modeling and convergent cross-mapping. https://ha0ye.github.io/rEDM/articles/rEDM.html#empirical-dynamic-modeling.

Appendix

All code used for this project may be found in my personal GitHub repository at: https://github.com/camiel1/Masters_Capstone. In addition, the *R* session info in conjunction with all activity for this Masters Capstone project is outlined below.

- R version 3.6.3 (2020-02-29), `x86_64-w64-mingw32`
- Running under: `Windows 10 x64 (build 17134)`
- Matrix products: default
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: clifro 3.2-2, dplyr 0.8.5, fable 0.1.2, fabletools 0.1.3, feasts 0.1.3, forcats 0.4.0, forecastML 0.9.0, fpp3 0.2, ggforce 0.3.1, ggplot2 3.3.0, ggridges 0.5.2, glmnet 4.0, gridExtra 2.3, lattice 0.20-38, lubridate 1.7.4, marmap 1.0.3, Matrix 1.2-18, ncdf4 1.17, parsnip 0.1.1, PerformanceAnalytics 2.0.4, purrr 0.3.3, quantmod 0.4-16, readr 1.3.1, rEDM 1.3.8.5, repr 1.1.0, rnoaa 0.9.5, seasonal 1.7.0, stringr 1.4.0, tibble 3.0.0, tidyquant 1.0.0, tidyr 1.0.0, tidyverse 1.2.1, timetk 1.0.0, tsibble 0.8.6, tsibbledata 0.1.0, TTR 0.23-6, xgboost 1.0.0.2, xts 0.12-0, yardstick 0.0.6, zoo 1.8-7
- Loaded via a namespace (and not attached): adehabitatMA 0.3.14, anytime 0.3.7, assertthat 0.2.1, backports 1.1.5, base64enc 0.1-3, bit 1.1-15.2, bit64 0.9-7, bitops 1.0-6, blob 1.2.1, broom 0.5.2, cellranger 1.1.0, class 7.3-15, cli 2.0.2, codetools 0.2-16, colorspace 1.4-1, compiler 3.6.3, crayon 1.3.4, crul 0.9.0, curl 4.3, data.table 1.12.8, DBI 1.1.0, digest 0.6.22, ellipsis 0.3.0, evaluate 0.14, fansi 0.4.0, farver 2.0.3, foreach 1.5.0, generics 0.0.2, glue 1.3.2, gower 0.2.1, grid 3.6.3, gtable 0.3.0, haven 2.1.1, hms 0.5.2, hoardr 0.5.2, htmltools 0.4.0, httpcode 0.2.0, httr 1.4.1, ipred 0.9-9, iterators 1.0.12, jsonlite 1.6, kableExtra 1.1.0, knitr 1.25,

lava 1.6.7, lifecycle 0.2.0, magrittr 1.5, MASS 7.3-51.5, memoise 1.1.0, modelr 0.1.5, munsell 0.5.0, nlme 3.1-144, nnet 7.3-12, pillar 1.4.3, pkgconfig 2.0.3, plyr 1.8.4, polyclip 1.10-0, pROC 1.16.2, prodlim 2019.11.13, quadprog 1.5-8, Quandl 2.10.0, R6 2.4.0, rappdirs 0.3.1, raster 3.0-12, RColorBrewer 1.1-2, Rcpp 1.0.2, RCurl 1.98-1.1, readxl 1.3.1, recipes 0.1.12, reshape2 1.4.3, rlang 0.4.5, rmarkdown 2.1, rpart 4.1-15, RSQLite 2.2.0, rstudioapi 0.11, rvest 0.3.5, scales 1.1.0, shape 1.4.4, sp 1.4-1, splines 3.6.3, stringi 1.4.3, survival 3.1-8, tidyselect 1.0.0, timeDate 3043.102, tools 3.6.3, tweenr 1.0.1, vctrs 0.2.4, viridisLite 0.3.0, webshot 0.5.2, withr 2.1.2, x13binary 1.1.39-2, xfun 0.10, XML 3.99-0.3, xml2 1.2.2, yaml 2.2.0