Data Science Project 2
Using Deep Learning to Predict Wave Heights in Domburg

For the Purpose of Surfing

By: Emil Abel Sigmann Engh

I. Introduction

The sport of surfing demands specific ocean and weather conditions. Relative to the standards of today, surfers in the past would invest a great amount of energy to find good surf conditions. The surf conditions were often unknown, and surfers searched for the best conditions at several spots. A revolution in the surfer lifestyle occured with the development of modern surf forecasting, as surf conditions at surf spots could be predicted and displayed on cellphones and computers. The troubles of finding good surf conditions was significantly reduced [1]. Furthermore, surf forecasting developed into a service industry, where companies predict surf conditions with numerical weather and wave models, and real-time data from offshore buoys. For wave prediction in present time, the commercial surf forecast website "Surfline" use the self-developed numerical wave model "Lola" [1]. The company "Magicseaweed" uses the numerical model "Wavewatch III"; developed by the National Oceanic and Atmospheric Association (NOAA) [2].

The challenge with modern numerical wave models is to accurately predict the surf height at the diverse surf breaks around the world. The complex relationship between open ocean wave height and subsequent surf height is inconsistent for each surf spot, and depends on shoreline features, the angle of surf break exposure, and the profile of the seafloor around the surf break (bathymetry). The worldwide prediction of surf heights by the means of a numerical model, would demand specific parameters on the features of each surf spot [3]. Such a numerical model would be computational expensive, and worldwide data on detailed surf spot features is currently unavailable [3] [4]. Therefore, machine learning is the focus of current development in surf height prediction, both in the industry and in academia [3] [4] [5]. A machine learning approach does not estimate parameters related to surf spot geographic features (shoreline features, angle of surf exposure and bathymetry) or local oceanic features (ocean currents and tides) but rather estimates the relationship between open ocean wave height and surf height by investigating patterns in data [5].

The commercial surf forecasts have shown to give inaccurate predictions of surf conditions in the province of Zeeland in the Netherlands. The local students and surf enthusiasts at University College Roosevelt often arrive at the most famous surfspot in Zeeland, called Domburg, with surf conditions different than predicted from the commercial forecasts. The reason for the inaccuracy might be due to the difference in weather conditions creating surf in the Netherlands in comparison to other, more known surf locations in the world. As waves leave a storm area, long-period waves have a higher velocity, and separate from shorter-period waves. The shorter-period waves quickly dissipate after leaving the wind-generating area, while longer period waves move forward in an organized wave-train called swell [6]. Generally, surf forecasts are developed to predict swell from faraway storm areas, as these weather and oceanic conditions determine standard surf conditions for most of the worlds surf zones. In contrast, standard surf conditions in the Netherlands often occurs from local storms close to shore, which creates shorter-period, yet surfable waves [7].

The purpose of this project is to develop a wave forecast model specifically for surf spots in Zeeland. The model will be data-driven, and made with Artifical Neural Networks (ANN). Three different ANN models will be developed, to investigate which method yields the most accurate result. An accurate wave forecast for Zeeland would be of great help to the surf enthusiasts at University College Roosevelt. A higher certainty of the local surf conditions would increase the amount of surfing in good conditions, and decrease the amount of sessions in poor conditions. More awareness of the local surf conditions would improve the time management of students, as they would know the correct time to prioritize surfing.

II. Methods

In this project, wind data from the North Sea will be used to predict wave heights in Domburg. Surfable waves at the Dutch coast are usually generated locally in the North Sea, due to the lack of exposure to open ocean swells. The locally generated waves have an average wavelength substantially shorter than open ocean swells, due to the enclosed geography of the North Sea; hindering waves to propagate far from their generation area. The longer-period waves lack space to separate from smaller period waves. Furthermore, long period waves are more rare in the North Sea than in the open ocean, due to the lack of length of water for the wind to blow over without obstruction (called fetch). An increase in fetch will increase the size and period of the waves [8]. As the surfable waves in the Netherlands mostly originate from local storms, local wind data will be used to predict the waves at Domburg. The predictors will be wind data from the past days, and from for the current time of wave observation.

A. Data Collection

The waves in the North Sea are well-suited to be measured with wave buoys. The spectral characteristics of wave buoys favor measurement of waves with short period. Furthermore, the North Sea is rather densely covered by wave buoys for atmospheric and oceanic measurement [9]. In Figure 1, one can see the location of the wave buoys that were used by Vorrips et al to create a wave forecast for the southern North Sea [9].

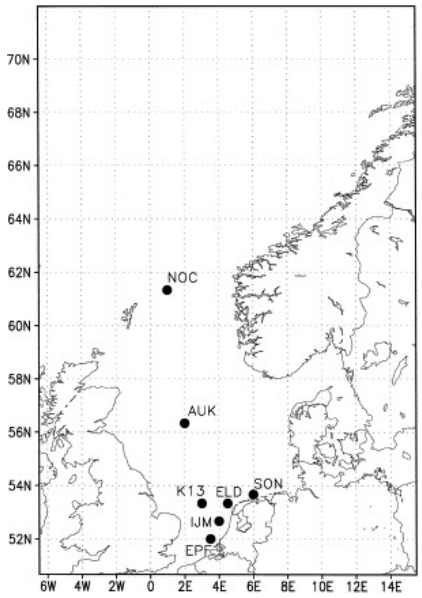


Figure 1: The location of wave buoys that were used for a wave forecast for the southern North Sea [9].

Wave buoys of similar location were used in this project. The wave buoy data was acquired from the website for the "Rijkswaterstaat" (Directorate-General for Public Works and Water Management in the Netherlands). The data was collected for the time period 1. January 2021 - 1. June 2021. Within this time period, observations occured at every tenth minute. The wind data concerns mean wind speeds (in m/s), and originate from three buoys located in the southern North Sea. The wave data concerns significant wave height (average wave height, from trough to crest, of the highest one-third of the waves) given in centimeters. The wave data would be the predicted variable in the model, and should represent the wave heigth at Domburg surf spot. Therefore, a wave buoy close to Domburg was chosen. Other relevant wave parameters for surf prediction is wave period, wave/wind direction and wind data in the northern North Sea [9]. Unfortunately, data on these parameters were unavailable. In Figure 2, one can see the location the buoys that were used in this project. The R code that was used to create the map in Figure 1 can be seen in the Appendix under subsection V-A.

Figure 2: Location of the wave buoys that were used in this project. The wave buoy contained wave height data, which would be the predicted variable in the model. The other buoys contained wind strength data which would be the predictors in the models.

B. Data Pre-processing

Both the simple model and advanced model would use current and past wind data to predict the waves at the buoy near Domburg. The simple model would use past wind data in the form of average values. For the past three days, the wind data was averaged for each day and each buoy. The advanced model used past wind data from every time step (ten minute timesteps throughout the day) for the past two days. The final dataset would then contain the current wind data from the three wind buoys, and the past wind data at these buoys for every ten minutes for the last 48 hours.

1) Pre-processing for the Simple Model: The wave dataset was joined with the wind dataset on the shared time-step variable (called "date_time" in the code). Then, nine new variables were added to each observation. The nine variables were the daily average wind strength for the three past days at the three buoys. The resulting dataset had twelve predictors: the wind strength at the three buoys for today, yesterday, two days ago and three days ago. Only a single variable was going to be predicted; the wave height at the wave buoy close to Domburg. The R code used for making the final dataset for the simple model can be seen in the Appendix under subsection V-B

2) Pre-processing for the Advanced Model: The wave dataset was joined with the wind dataset on the shared time-step variable (called "date_time" in the code). Then, for each observation, wind data from each wind buoy was added for every tenth minute during the last 48 hours. Each wave observation had 867 predictors: the current wind data from the three buoys for today, and every tenth minute for the past 48 hours. The advanced model did not include wind data up to 72 hours (three days), like the simple model did, due to long and inconvenient computation times. The R-code used for making the final dataset for the advanced model can be seen in the Appendix under subsection V-C.

C. Creating the twoANN models

The process of developing the ANN networks was inspired by the example of an ANN problem in subsection 6.3 of the academic book "Deep Learning With R" by Challet & Allaire. The problem is fairly similar to the problem in this project, as a weather parameter (temperature) is predicted by the means of other weather parameters (air temperature,humidity, wind direction etc,) from the past. However, the problem from Challet & Allaire has some differences from this project. The predictors in the problem from Challet & Allaire have different scales, and need to be normalized. Furthermore, only predictors of the past are used to predict current temperature [10]. In this project, the predictors have the same scale (wind speed parameters are the only predictors in this problem), and the predictors occur at the same time as the response, and in the past.

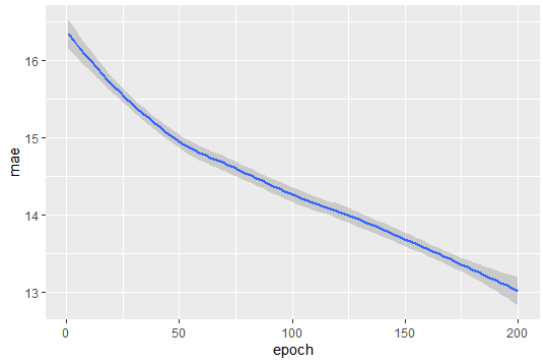
In both ANN models, two hidden layers with linear activation was used. Furthermore, four-fold cross-validation was used to investigate the suitable amount of units for each layer, and the optimal amount of epochs. Also, cross-validation would indicate if the models were overfitting the training data.

1) Simple ANN model: Four-fold cross validation was used to find the combination of units in the two hidden layers, and the amount of epochs, that give the best performance. The network was trained on the analysis data with 200 epochs. In Table I, one can see the combination of units in the two dense layers, and the corresponding lowest mean absolute error (mae) on the validation data. The mae is averaged over the four folds. The lowest mean absolute error was obtained when the first and second dense layer had 100 units respectively. The network was not trained with two dense layers with higher than 100 units, due to long computational times.

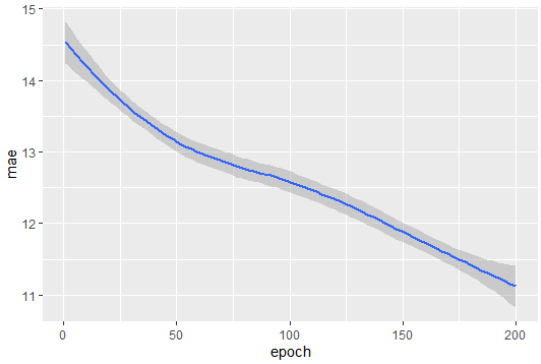
Unit 1	Unit 2	Lowest mae
50	50	12.83
100	50	10.68
100	100	9.79

Table I: The amount of units in the first layer (Unit1), and second layer (Unit2), and the corresponding average lowest mae over the four folds.

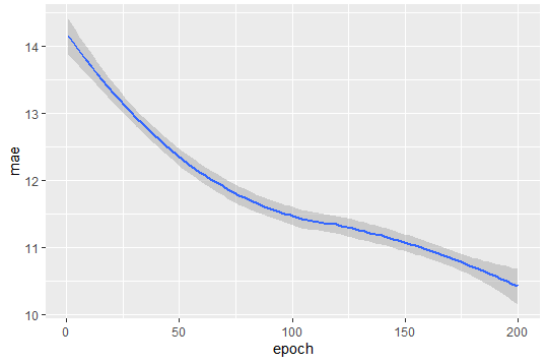
In Figure 3, the average mae for the four folds is plotted for the amount of epochs that have been run. The three sub figures represent the three networks with different combination of units in each layer. The combination of units in each network can be seen in Table I. As one can see in Figure 3, performance was highest for the three networks when 200 epochs was used. The network was not trained for a higher amount of epochs due to low computational times. The R code for the development of the simple model can be seen in the Appendix under subsection V-D



(a) Network with 50 units in the two layers.



(b) Network with 100 units in the first layer, and 50 units in the second layer.



(c) Network with 100 units in the first layer, and 100 units in the second layer.

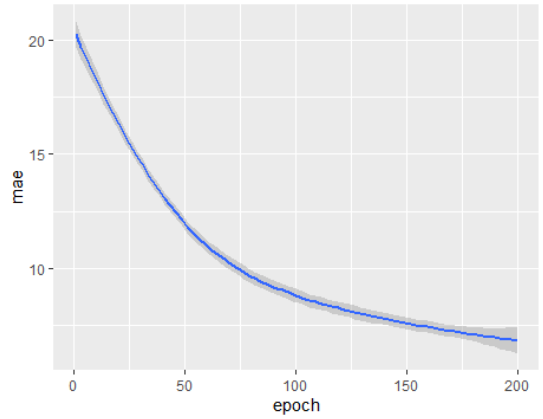
Figure 3: The average mae plotted against the number of epochs for each network.

2) Advanced ANN model: Four-fold cross validation was used to find the combination of units in the two hidden layers, and the amount of epochs, that give the best performance. The network was trained on the analysis data with 200 epochs. In Table II, one can see the combination of units in the two dense layers, and the corresponding lowest mean absolute error (mae) on the validation data. The mae is averaged over the four folds. The lowest mean absolute error was obtained when the first and second dense layer had 100 units respectively. The network was not trained with two dense layers with higher than 100 units, due to long computational times.

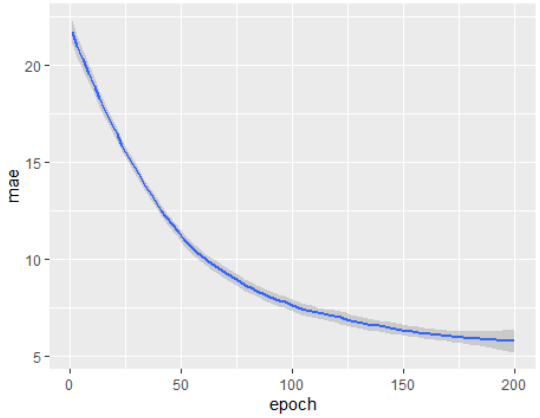
Unit 1	Unit 2	Lowest mae
50	50	5.5
100	50	4.59
100	100	4.22

Table II: The amount of units in the first layer (Unit1), and second layer (Unit2), and the corresponding average lowest mae over the four folds

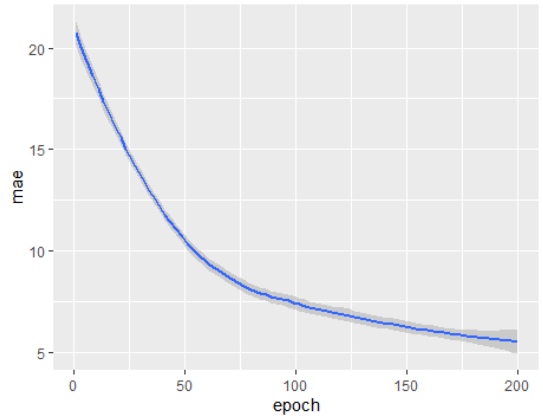
In Figure 4, the average mae for the four folds is plotted for the amount of epochs that have been run. The three sub figures represent the three networks with different combination of units in each layer. The combination of units in each network can be seen in Table II. As one can see Figure 4, performance was highest for the three networks when 200 epochs was used. The network was not trained for a higher amount of epochs due to low computational times. The R code for the development of the advanced model can be seen in the Appendix under subsection V-E



(a) Network with 50 units in the two layers.



(b) Network with 100 units in the first layer, and 50 units in the second layer.



(c) Network with 100 units in the first layer, and 100 units in the second layer.

Figure 4: The average mae plotted against the number of epochs for each network.

D. Testing the advanced model

The advanced model that ran on 200 epochs and contained 100 units in the two hidden layers had the highest performance during cross validation. Therefore, this model was fitted on the test set. The mae of the total test set was computed. The aim of the project is to predict good surfing conditions. Therefore, its important to investigate the models prediction on good surf days. Often, good surf conditions in Domburg demand wave heights of 100cm. Therefore, the test set was filtered to only contain wave observations above 100cm. Then, the mae of the predicted waves above 100cm was computed. The R code used for fitting the model on the test set and calculating the mean absolute errors can be seen in the Appendix under subsection V-F.

III. Results and Discussion

A. Testing of the advanced model

The advanced model running on 200 epochs and containing 100 units in the two hidden layers was fitted on the test data. The test mae was 5,622 on the entire test set. When the test set was filtered to only contain wave heights above 100cm, the test mae was 9,09. Therefore, one can can expect the forecast to have an error of 9cm when waves above 100cm is predicted. If the surf report made in this project predicts waves of 100cm, one can expect the surf to be in the range of 91 - 109cm. For the purposes of this project, the result is satisfactory. Students at University College Roosevelt would have had an improved confidence of the incoming wave heights with this model, and accurate wind data inputs.

One can get a better understanding of the model performance by comparing the model with the random chance of getting surf above 100cm. The test dataset was constructed by randomly selecting 25% of the observations from the advanced dataset created in the pre-processing of the advanced model (subsection II-B2). Based on this test set, one is 30% likely to go surfing in waves higher than 100cm, if one goes surfing at a random time. If one use the advanced model made in this project, and go surfing on a day where the waves are predicted to be higher than 100cm, the probability of the waves to be over 100cm in

reality is around 96%. From this example, it is clear that the surf forecast significantly improves your chances to go surfing when conditions are good. The probability calculations in this example were coded in R, and can be seen in the Appendix under subsection V-G

B. Improvements that can be made

Wave period is an important factor that depicts surf quality. If the wave height is 100cm, the surf quality is vastly different for wave periods of four or ten seconds. In general, longer period waves contain more energy, and give better surf conditions. For an accurate surf prediction, wave period should be considered. Unfortunately, data on wave period was not available from the buoy data obtained in this project. To improve the confidence of the surf forecast, wave period should be included in the future. However, data on wave periods in the North Sea would need to be available for the same timesteps as the wave and wind data.

The forecast model created in this project does not consider wind/wave direction. Therefore, the model might predict waves that move away from shore. However, this is highly unlikely to occur when the model is predicting wave heights that are surfable. The buoy that was used to obtain wave data in this project is located close to the shore. Even very strong offshore winds are unlikely to generate surfable wave heights at the buoy in the offshore direction, due to the low fetch between the buoy and the shore. Regardless of this fact, wave direction is beneficial to include in a surf forecast for Domburg, as conditions might vary depending on the angle of the incoming waves. Waves coming from a westerly angle tend to create dominant leftgoing waves, while waves coming from a north-easterly angle creates dominant rightgoing waves.

The model depends on precise weather forecasts for the accurate predictions of wave heights. The current advanced model demands 867 inputs of wave data from the last 48 hours from three different wave buoys. Some of the predictors (wind strength) occur at the same time as the response (wave height). Therefore, these predictors need to be acquired from weather forecasts, which may reduce the accuracy of the model. Future focus on the model should consider predicting wave heights with wind forecast input data, and then investigate the accuracy of the prediction. Also, future work could focus on only using predictors that can be recorded (e.g., wind data from 24 hours prior to wave prediction) while still maintaining a high accuracy. Also, the accuracy of such a model could be compared with the accuracy of the model made in this project, when forecasted wind data is used as inputs.

IV. Conclusion

The aim of this project was to create a surf forecast specifically for Domburg surf spot in Zeeland, Netherlands. The local students at University College Roosevelt would benefit from having a more accurate prediction for the coming surf. The forecast was created by developing a neural network that predicts wave heights offshore. The predictors of the model was wind data from three offshore buoys in the North Sea. Two neural network models were created: a ”simple” network with 12 predictors, and a ”advanced” network with 867 predictors. The advanced model produced more accurate predictions. Therefore, this model evaluated on the the testing data. The result of the model testing was satisfactory, meaning the model would be helpful for students at University College Roosevelt to be better informed of wave heights coming in the near future. However, the model has some shortcomings. First of all, the model might be difficult to implement, due to the large amount of input data that is needed. Also, some of the input data concerning wind strength should occur at the same time as the predicted waves, and needs to be forecasted. Last of all, the model does not predict wave direction or wave period, which are both important parameters for good surf conditions, due to lack of accessible data.

V. Appendix

A. Mapping

The code below was used to map the location of the buoys from where data was collected. The map showing the buoy locations can be seen in Figure 1 in subsection II-A.

```
options(digits = 15)
 wave_coord <- domburg_waves_upload %>%
   select(MEETPUNT_IDENTIFICATIE, X, Y) %>%
   rename("buoy_name" = MEETPUNT_IDENTIFICATIE) %>%
   group_by(X, Y, buoy_name) %>%
   summarise() %>%
   ungroup() %>%
   mutate(
     buoy_type = "wave buoy"
   )
 lonlat <- wave_coord %>%
   select(X, Y) %>%
   group_by(X, Y) %>%
   summarise() %>%
   mutate(
     X = as.numeric(gsub(",",".", X)),
     Y = as.numeric(gsub(",",".", Y))
   ) %>%
   rename(
     "lon" = X,
     "lat" = Y
   )
 v <- vect(lonlat, crs="+proj=utm +zone=31 +datum=WGS84  +units=m")
 y <- project(v, "+proj=longlat +datum=WGS84")
 lonlat <- geom(y) %>%
   as_tibble() %>%
   select(x, y)
 wave_coord <- wave_coord %>%
   bind_cols(lonlat) %>%
   select(buoy_name, buoy_type, x, y)
 wind_coord <- wind_data_upload %>%
   select(MEETPUNT_IDENTIFICATIE, X, Y) %>%
   rename("buoy_name" = MEETPUNT_IDENTIFICATIE) %>%
   group_by(X, Y, buoy_name) %>%
   summarise() %>%
   ungroup() %>%
   mutate(
     buoy_type = "wind buoy"
   )
 lonlat <- wind_coord %>%
   select(X, Y) %>%
   group_by(X, Y) %>%
   summarise() %>%
   mutate(
     X = as.numeric(gsub(",",".", X)),
     Y = as.numeric(gsub(",",".", Y))
   ) %>%
   rename(
     "lon" = X,
     "lat" = Y
   )
 v <- vect(lonlat, crs="+proj=utm +zone=31 +datum=WGS84  +units=m")
 y <- project(v, "+proj=longlat +datum=WGS84")
```

```
lonlat <- geom(y) %>%
  as_tibble() %>%
  select(x, y)
wind_coord <- wind_coord %>%
  bind_cols(lonlat) %>%
  select(buoy_name, buoy_type, x, y)
Final_coord <- wind_coord %>%
  bind_rows(wave_coord)
Map <- leaflet() %>%
  addPopups(data = Final_coord, lat = ~y, lng = ~x, popup = ~buoy_type, options = popupOptio
  addTiles() %>%
  setView(lng = 3.4, lat = 52.5, zoom = 7.45)
Map
```

B. Data pre-processing for the simple model

The code below was used to pre-process the data used for the simple nerual network model.

```
domburg_waves <- domburg_waves_upload %>%
  select(WAARNEMINGDATUM, REFERENTIE, GROOTHEID_OMSCHRIJVING, NUMERIEKEWAARDE) %>%
  unite("date_time", WAARNEMINGDATUM, REFERENTIE, sep = " ") %>%
  mutate(
    date_time = parse_date_time(date_time, "%d%m%Y %H%M%S"),
    NUMERIEKEWAARDE = as.numeric(gsub(",",".", NUMERIEKEWAARDE))
  ) %>%
  pivot_wider(names_from = GROOTHEID_OMSCHRIJVING, values_from = NUMERIEKEWAARDE) %>%
  rename(
    domburg_wave_height = `Significante golfhoogte in het spectrale domein`
  ) %>%
  filter(
    nchar(domburg_wave_height) <= 3
  )
wind_wrangled <- wind_data_upload %>%
  select(MEETPUNT_IDENTIFICATIE, WAARNEMINGDATUM, REFERENTIE,NUMERIEKEWAARDE) %>%
  unite("date_time", WAARNEMINGDATUM, REFERENTIE, sep = " ") %>%
  mutate(
    date_time = parse_date_time(date_time, "%d%m%Y %H%M%S"),
    measuring_point = MEETPUNT_IDENTIFICATIE,
    wind_strength = as.numeric(gsub(",",".", NUMERIEKEWAARDE))
  ) %>%
  filter(nchar(wind_strength) <= 5) %>%
  select(-c(NUMERIEKEWAARDE, MEETPUNT_IDENTIFICATIE)) %>%
  pivot_wider(names_from = measuring_point, values_from = wind_strength) %>%
  mutate(
   one_day_past_date = as.Date(date_time) - ddays(1),
   two_day_past_date = as.Date(date_time) - ddays(2)
  )
avg_daily_wind <- wind_wrangled %>%
  mutate(
    date = as.Date(date_time)
  ) %>%
  group_by(date) %>%
  summarise(
    avg_Europlatform = mean(Europlatform),
    avg_K13_alpha = mean(`K13 Alpha`),
    avg_Q1_platform = mean(`Q1 platform`)
  )
dataset
```

```
wind_final <- wind_wrangled %>%
  inner_join(avg_daily_wind, by = c("one_day_past_date" = "date")) %>%
  rename(
    "1day_past_avg_Europlatform" = avg_Europlatform,
    "1day_past_avg_K13_alpha" = avg_K13_alpha,
    "1day_past_avg_Q1_platform" = avg_Q1_platform
  ) %>%
  inner_join(avg_daily_wind, by = c("two_day_past_date" = "date")) %>%
  rename(
    "2day_past_avg_Europlatform" = avg_Europlatform,
    "2day_past_avg_K13_alpha" = avg_K13_alpha,
    "2day_past_avg_Q1_platform" = avg_Q1_platform
  ) %>%
  select(-c(one_day_past_date, two_day_past_date))
wind_wave <- domburg_waves %>%
inner_join(wind_final, by = "date_time") %>%
na.omit(
```

C. *Data pre-processing for the advanced model*

The code below was used to pre-process the data used for the advanced neural network model.

```
wave_wind_current <- domburg_waves %>%
  inner_join(wind_wrangled, by = "date_time") %>%
  select(-c(one_day_past_date, two_day_past_date))
wave_wind_current <- wave_wind_current[rev(order(wave_wind_current$date_time)),]
wave_wind_current <-  wave_wind_current %>%
  mutate(index = row_number())
total_rows <- tibble()
for (x in 1:864) {
  total_rows[x] <- as.numeric("")
  colnames(total_rows)[x] <- paste(x)
}
total_rows <- total_rows %>%
  mutate(index = as.numeric(""))
for (i in 1:nrow(wave_wind_current)) {
  # 72 hrs
  cut_min <- wave_wind_current[i+1,]$index
  cut_max <- wave_wind_current %>%
    filter(date_time == wave_wind_current[i,]$date_time - ddays(2))
  if (nrow(cut_max) == 0) {
    total_rows <- total_rows
  }
  else {
    cut_max <- cut_max$index
    cut_rows <- wave_wind_current %>%
      slice(cut_min:cut_max) %>%
      select(-c(index, domburg_wave_height))
    if (nrow(cut_rows) < 288) {
      total_rows <- total_rows
    }
    else {
      cut_rows <- cut_rows %>%
        pivot_wider(names_from = date_time, values_from = c("Europlatform", "K13 Alpha", "Q1
        mutate(index = i)
      #for (y in 1:ncol(cut_rows)) {
       # colnames(cut_rows)[y] <- paste(y)
      #}
```

```
        colnames(cut_rows) <- colnames(total_rows)
        total_rows <- total_rows %>%
          bind_rows(cut_rows)
      }
    }
  }
 final wave-wind dataset
final_set <- wave_wind_current %>%
  inner_join(total_rows, by = "index") %>%
  select(-c(date_time, index)) %>%
  na.omit()
```

D. Simple ANN model

The code below was used create the simple ANN model

```
data <- initial_split(wind_wave, prop = 3/4)
 training <- training(data)
 x1 <- training %>%
   select(Europlatform, `K13 Alpha`, `Q1 platform`, `1day_past_avg_Europlatform`,
          `1day_past_avg_K13_alpha`, `1day_past_avg_Q1_platform`, `2day_past_avg_Europlatform
          `2day_past_avg_K13_alpha`, `2day_past_avg_Q1_platform`) %>%
   as_tibble()
 y1 <- training %>%
  pull(domburg_wave_height)
 testing <- testing(data)
 x2 <- testing %>%
   select(Europlatform, `K13 Alpha`, `Q1 platform`, `1day_past_avg_Europlatform`,
          `1day_past_avg_K13_alpha`, `1day_past_avg_Q1_platform`, `2day_past_avg_Europlatform
          `2day_past_avg_K13_alpha`, `2day_past_avg_Q1_platform`) %>%
   as.matrix()
 y2 <- testing %>%
   pull(domburg_wave_height)
 r <- recipe(~., data = x1) %>%
   prep()
 cross_v <- vfold_cv(training, v = 4)
 model <-  keras_model_sequential() %>%
   layer_dense(units = 100, activation = "relu") %>%
   layer_dense(units = 100, activation = "relu") %>%
   layer_dense(units = 1)
 model %>%
   compile(
     loss = "mse",
     optimizer = "rmsprop",
     metrics = c("mae")
   )
 fitdm <- function(split) {
   xan <- r %>% bake(new_data = analysis(split)  , composition = "matrix")
   xas <- r %>% bake(new_data = assessment(split), composition = "matrix")
   yan <- analysis(split)  %>% pull(domburg_wave_height)
   yas <- assessment(split) %>% pull(domburg_wave_height)
   history <- model %>%
     fit(x = xan, y = yan, validation_data = list(xas, yas), epochs = 200, batch_size = 125)
   history %>%
     as_tibble()
 }
 res <- cross_v %>%
   mutate(mod = map(splits, fitdm)) %>%
```

```
  unnest(mod)
highest <- res %>% filter(metric == "mae", data == "validation") %>%
  group_by(epoch) %>%
  summarize(mae = mean(value))
highest %>%
  ggplot(aes(epoch, mae)) + geom_smooth()
```

E. Advanced ANN model

The code below was used to develop the advanced ANN model.

```
data2 <- initial_split(final_set, prop = 3/4)
training2 <- training(data2)
x1 <- training2 %>%
  select(-c(domburg_wave_height)) %>%
  as.matrix()
y1 <- training2 %>%
  pull(domburg_wave_height)
testing2 <- testing(data2)
x2 <- testing2 %>%
  select(-c(domburg_wave_height)) %>%
  as.matrix()
y2 <- testing2 %>%
  pull(domburg_wave_height)
r <- recipe(~., data = x1) %>%
  prep()
cross_v <- vfold_cv(training2, v = 4)
model2 <- keras_model_sequential() %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 100,activation = "relu") %>%
  layer_dense(units = 1)
model2 %>%
  compile(
    loss = "mse",
    optimizer = "rmsprop",
    metrics = c("mae")
  )
fitdm <- function(split) {
  xan <- r %>% bake(new_data = analysis(split)  , composition = "matrix")
  xas <- r %>% bake(new_data = assessment(split), composition = "matrix")
  yan <- analysis(split)  %>% pull(domburg_wave_height)
  yas <- assessment(split) %>% pull(domburg_wave_height)
  history <- model2 %>%
    fit(x = xan, y = yan, validation_data = list(xas, yas), epochs = 200, batch_size = 125)
  history %>%
    as_tibble()
}
res <- cross_v %>%
  mutate(mod = map(splits, fitdm)) %>%
  unnest(mod)
highest <- res %>% filter(metric == "mae", data == "validation") %>%
  group_by(epoch) %>%
  summarize(mae = mean(value))
highest %>%
  ggplot(aes(epoch, mae)) + geom_smooth()
```

F. Testing the advanced ANN model

Below is the code used to evaluate the advanced ANN model on the test set, and the accuracy of the ANN model for wave heights above 100cm.

```
train_fit <- model2 %>%
    fit(x = x1, y = y1, epochs = 200, batch_size = 125)
  model2 %>% evaluate(x2, y2)
  model2 %>% keras::get_weights()
  pred <- model2 %>%
    predict_on_batch(x2) %>%
    as_tibble() %>%
    rename("pred" = V1) %>%
    mutate(index = row_number())
  higher_wave_pred <- testing2 %>%
    mutate(index = row_number()) %>%
    inner_join(pred, by = "index") %>%
    select(domburg_wave_height, pred) %>%
    filter(domburg_wave_height >= 100)
  higher_wave_pred %>%
    mae(domburg_wave_height, pred)
```

G. Probability Calculations

Below is the code to calculate the probability of surf above 100cm when the advanced ANN model predicts waves above 100cm. Also, the probability to experience 100cm high waves or above when one randomly goes for a surf session is showed below.

```
higher_wave_pred %>%
    filter(domburg_wave_height >= 100) %>%
    nrow()
  higher_wave_pred %>%
    filter(pred < 100)
  1 - 20/729
  testing2 %>%
    nrow()
  testing2 %>%
    filter(domburg_wave_height >= 100) %>% nrow()
  729/2427
```

REFERENCES

[1] S. Thornton, "Science of surfline," *National Geographic*, 2012. [Online]. Available: https://www.nationalgeographic.org/article/surfs/

[2] "Surf forecast accuracy," *Magicseaweed*, 2007. [Online]. Available: https://magicseaweed.com/news/surf-forecast-accuracy/4769/

[3] M. Carney, P. Cunningham, J. Dowling, and C. Lee, "Predicting probability distributions for surf height using an ensemble of mixture density networks," *Proceedings of the 22nd international conference on Machine learning*, pp. 113 – 120, 2005.

[4] S. C. James, Y. Zhang, and F. O'Donncha, "A machine learning framework to forecast wave conditions," *Coastal Engineering*, vol. 137, p. 1–10, 2018.

[5] B. Freeston, "Machine learning for surf forecasting," *Surfline Labs*, 2018. [Online]. Available: https://medium.com/surfline-labs/machine-learning-for-surf-forecasting-4a007f13b3e3#:~:text=Surf%20forecasting%20is%20really%20a,together%20the%20correlations%20and%20causations.&text=This%20is%20what%20machine%20learning,in%20huge%2C%20complex%20data%20sets.

[6] "Wave energy, decay and direction," *Surfline*, 2017. [Online]. Available: https://www.surfline.com/surf-news/wave-energy-decay-direction/2445

[7] P. Tigges, "Learning how to forecast waves in 10 minutes." *Pepijn Tigges*. [Online]. Available: https://www.iampepijn.nl/blog/learn-to-forecast-waves

[8] A. Semedo, R. Vettor, Øyvind Breivik, A. Sterl, M. Reistad, C. G. Soares, and D. Lima, "The wind sea and swell waves climate in the nordic seas," *Ocean Dynamics*, vol. 65, p. 223–240, 2015.

[9] A. C. Vorrips, "Spectral wave data assimilation for the prediction of waves in the north sea," *Coastal Engineering*, vol. 37, no. 3-4, pp. 455–469, 1999.

[10] F. Chollet and J. J. Allaire, *Deep Learning With R*. Manning Publications, 2017.