Common Eider Migration Pattern

COMP 551: Applied Machine Learning
Final Project Report

Corbin Hopper
corbin.hopper@mail.mcgill.ca
260577638

Yue Xia
yue.xia@mail.mcgill.ca
260556275

Hoai Phuoc (HP) Truong
phuoc.truong2@mail.mcgill.ca
260526454

I. Introduction

Understanding bird migration is crucial to managing how humans interfere with their habitats. Conservation must consider both where birds currently reside and where they will be in the near future. Machine learning is one solution that can leverage publically available datasets with minimal cost and intervention. Tracking data of common eiders (Somateria mollissima) is used to predict the the eider's future location. Past movements are combined with environmental factors to form robust models of eider movement. Models are trained on a classification and a regression task. In the former task the data is first clustered into a discrete set of locations, whereas the later groups data by time intervals instead.

The recurrent neural network performes the best for classification, implying that the complexity of the classification task and the network architecture is appropriate for the problem at hand. Meanwhile, for regression the logistic regression models outperforms a recurrent neural network. The amount of data is likely insufficient for the complex neural architecture to capture the bird's precise location. The coarser logistic model also reveales the relevance of various features. Soil and snow temperature are shown to provide more useful information than month of year, suggesting that climate change will alter eider migration time. Our models are able to predict the location of a common eider in the next few days with an error of 74 kilometers, and its location in the next month with an error of 198 kilometers. These are feasible temporal and spatial ranges to restrict invasive human activities, such as fishing, in eider habitats.

II. Related works

Tracking and identifying animal movement is costly and often harms the animals themselves [8]. Machine learning is capable of assessing animal movements without nonstop use of geolocators. Tracking data provides a wide variety of possible problems. Most works adopt a behaviorist paradigm by classifying animal behavior in order to better understand them [5, 6, 11]. Other approaches use semi-supervised learning to label animal movement [1]. In contrast to the last two, our approach focuses on predicting future animal movement. Conservation necessitates management of human activity in relation to animal habitat and so requires a robust understanding of animal movement. Moreover, an understanding of the features that drive animal movement will be vital to understanding how their movement changes over time, especially in response to our rapidly changing environment.

III. Data set

The models are built around an extended population of common eiders residing in Alaska, far eastern Russia, and small pieces of Northern Canada. The data is acquired from MoveBank [12] [10], a publicly available database on animal movement across the globe. Common eiders are migratory seabirds with close kinship ties. Migration occurs in flocks and females can even care for others' young [7]. Common eiders feed on mussels and other small sea organisms. As a result they are highly affected by overfishing, and multiple studies have linked extensive fishing to rapid decline in eider populations [2, 3]. If eider movement can be predicted, fisheries can be preemptively restrained prior to eider arrival.

The data set consists of 110 common eiders, each with 300 to 700 measurements taken within a year for each individual. These culminate in 65555 raw data points. Each measurement records the longitude and latitude of the common eider at a time stamp. We also collect the environmental data at that time and position, which includes sunshine duration, visibility, sea surface roughness, wind velocity, air pressure, atmospheric water temperature, snow temperature, ice temperature, soil temperature, distance to the coast, runoff and soil water. A preliminary plot of the positions of the bird can be seen in figure 1 below.

IV. Target prediction tasks

A. Predicting exact location

Our first approach to forming a task is predicting the exact location (i.e. longitude and latitude) of a bird given its positions and features in the previous few time periods. The duration of the time period will be 4 days, 7 days (1 week) and 30 days (1 month). For each duration listed, we **consider three continuous time periods and predict the position of the bird in the fourth one**. The choice of the durations are based on the amount of possible samples obtained at the end of the grouping process. In particular, choosing to group our data for every 4

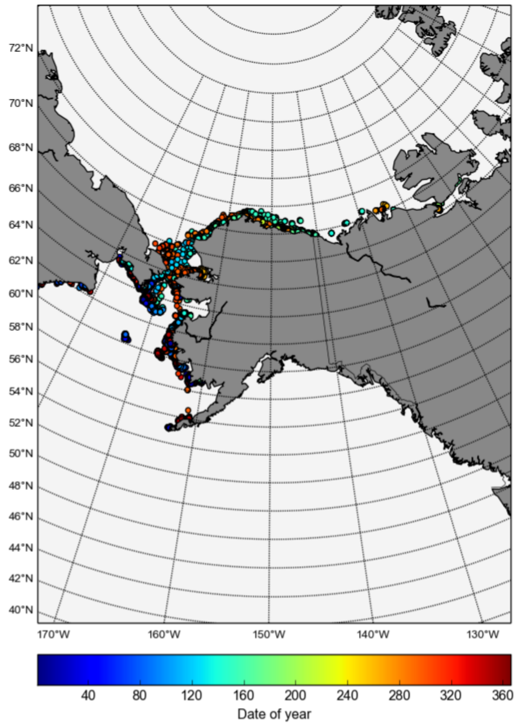

Figure 1: Birds' positions by date

days, 7 days and 30 days gave the most amount of continuous time periods that can be used for training the models. We consider all features provided by combining features from the original data set and the obtained environmental data. for all three time periods in our prediction model. The performance of the prediction models are measured by calculating the Coefficient of Determination (R^2 score). The characteristic and interpretation of R^2 score will be discussed in more details in the algorithm design and result section of the report. This score is calculated as follow:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1}$$

B. Predicting by clusters

Another approach we chose to explore is to first identify clusters using the positions of the birds, and then predict which cluster the birds will be in given its past information. K-means clustering is used to assign clusters to each data point given their position. The number of clusters is determined from a combination of quantitative and qualitative methods. A graph displaying the reduction in error is used to assess when more clusters did not significantly improve accuracy. This resultes in a small range of k, but not an exact value. Clusters are then visualized on a map to intuitively confirm their relevance to the problem at hand. An agglomerative method based on ant-colony clustering is also attempted, where similar datapoints are grouped together in a bottom-up manner. However, long run times coupled with multiple redundant clusters renders it ineffective.[4]

V. Data pipeline

Longitude and latitude is a geographic coordinate system to effectively identify a location on Earth's surface. This coordinate system is extremely popular in specifying global locations. However, coordinates (pair of longitude-latitude) in this system do not obey several fundamental properties that are commonly assumed during data analysis. Due to the spherical nature of Earth, several notable examples of abnormality in longitude-latitude coordinate system are mean calculation of a set of coordinates in this system, and distance calculation between two points on this coordinate. To ease the data processing task in this project, we assume Earth to be a perfect sphere and convert any given longitude and latitude to its 3D representation (x, y, and z) using the following transformation after conversion to radians (ignoring the radius of Earth since it is constant). This classic representation is more manageable as conventional arithmetic operations can be applied on coordinates in this form. Acknowledging that the prediction results will be in its 3D representations (x, y, and z), the R^2 score will be calculated in such representation, as opposed to the longitude - latitude representation. The conversion between the two coordinate systems is as follows:

$$\begin{cases} x = cos(latitude) * cos(longitude) \\ y = cos(latitude) * sin(longitude) \\ z = sin(latitude) \end{cases} \tag{2}$$

Our data pipeline is described in the following figure:

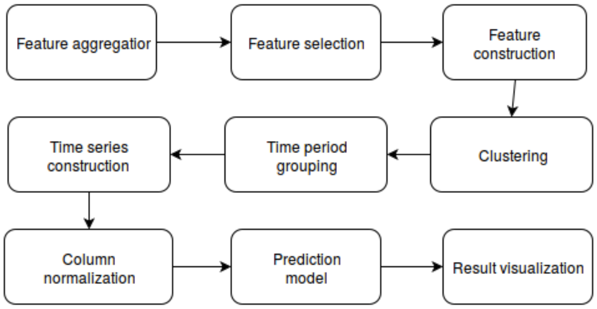

Figure 2: General data pipeline

1) **Feature aggregation**: aggregate features from original data set and pair with the obtained ENV data. At the end of the feature aggregation stage, we extracted 52 features.
2) **Feature selection**: manual selection of relevant features. In this step, we investigate the meta data of each feature and eliminate features which have any of the following traits: missing more than 80% of the data (value is *nan*), float columns which have under 50 distinct values (out of 65555 samples), specific human friendly (but not very useful) features (e.g. bird name), and repeated columns (since the ENV data contains columns that repeat the

information in the original data set under a different name). Using this approach, we eliminate 23 irrelevant features from the aggregated data set.

3) **Feature construction**: convert longitudes and latitudes to their 3D representation using the formulae provided in the beginning of this section. This stage replaces the longitude and latitude feature by three new features: X, Y, and Z.
4) **Clustering**: The clustering task requires extra preprocessing. A clustering algorithm identifies clusters of positions and assigns each corresponding datapoint to one such cluster.
5) **Time period grouping**: using the timestamp provided in each row, group rows that are in the same time period (e.g. if period is 7 days then combine all rows that are in the same week since the beginning of the study together). This shrinks our input data set depending on the chosen duration of the time period. In particular, grouping by 4 days, 7 days and 30 days result in 7259, 4744, and 1316 combined rows respectively.
6) **Time series construction**: construct a time series using 4 consecutive time periods. The first 3 continuous time periods are concatenated by their feature columns in order (e.g. if one period is 4 days then construct a series whose length is 12 days). Their desired output value is the location at the *next* time period. If one of the 4 consecutive time periods is missing data, the series is not included.
7) **Column normalization**: normalize feature columns using rescaling ($X' = \frac{x-min}{max-min}$). We chose this over standardization ($X' = \frac{X-\mu}{\sigma}$) because most observed features do not have a Gaussian distribution. For instance, snow temperature will often be 0 when there is not snow, tending towards a bimodal distrubtion with two means for snow and no snow.
8) **Prediction model**: different machine learning models predict the location of the bird. These models will be discussed in details in section VI of the report.
9) **Result visualization**: in addition to evaluating the performance of the model based on conventional metrics (e.g. R^2 score, $0-1$ accuracy), we visualize the prediction on the world map to better understand the obtained results. For the task of predicting the exact location of the birds, outputs are sets of x, y, and z coordinates and are converted back to longitudes and latitudes to visualize them on a map.

VI. Algorithm Selection and Implementation

A. Predicting exact location

1) Linear models: For all linear models, we set aside 10% of our data as a test and use k-fold cross validation on the rest to train the data with $k = 10$ folds.

a) Linear Regression: A standard linear regression implementation from scikit-learn [9] to train the input data.

b) Lasso Regression: A standard Lasso regression implementation from scikit-learn [9] to train the input data with $iteration\_count = 2000$ and various values of $\alpha \in \{0.1, 0.00001\}$.

c) Ridge Regression: A standard Ridge regression implementation from scikit-learn [9] to train the input data with various values of $\alpha \in \{0.1, 0.01\}$, $tolerance\_threshold = 0.00001$ and no limit on $iteration\_count$.

2) RNN: The recurrent neural network representation is shown in Figure 3. Multiple series of positions along with the normalized environmental measurements at the corresponding positions are used as inputs. Each position is represented by x, y, and z coordinates transformed from longitudes and latitudes. Meanwhile the outputs are only series of positions, represented in x, y, z coordinates.

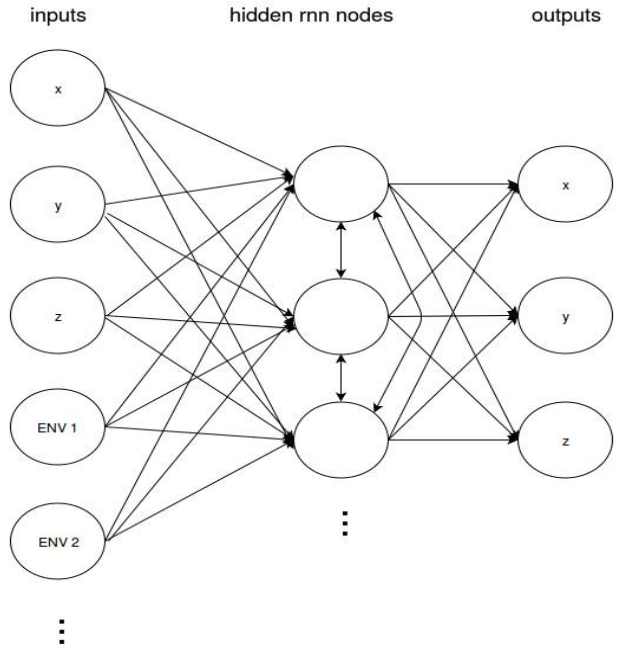

Figure 3: The recurrent neural network

In terms of preliminary design decisions, all the hidden nodes are inter-connected and activated via the hyperbolic tangent sigmoid function. Output units apply a linear function on the results of hidden nodes. At initialization all the weights and biases feeding into each neuron are normalized according to uniform distributions between 0 and 1. The optimized loss functions is the sum of squares cost function, which represents the squared distances between two points in 3-D. The model is trained by stochastic gradient descent with an adapted learning rate initialized at `0.0001` and halved when the testing error increases. The error is measured every 5 gradient descend steps. Finally, to prevent over-training, the model is tested on a validation set every 5 gradient descend steps, and the training stops if the validation accuracy keeps decreasing for 3 consecutive evaluations.

The number of parameters are tuned as a hyper-parameter. Models with 10, 25, 50 and 100 hidden units are trained with the grouping by month data over 200 epoches. The results show that 25 neurons give the best accuracy, as shown in Table I. Since the training takes a very long time, the number of hidden units is fixed at 25 for grouping by any time period without performing separate tests.

Number of hidden units	10	25	50	100
Validation accuracy	0.0116	0.0084	0.0098	0.0110

Table I: errors in predicting exact positions with RNN

B. *Predicting by clusters*

1) Logistic Regression: A standard implementation of Logistic Regression from scikit-learn [9] is used. We try out with different regularization strengths $C \in \{1.0, 0.5, 0.1\}$ where smaller C indicates stronger regularization strength.

2) Random forest: A standard implementation of Random Forest from scikit-learn [9] is used. We try out with number of trees $m \in \{50, 100\}$.

3) RNN: In the classification task, the positions in both inputs and outputs are represented by 1-hot encodings of the cluster numbers. The structure for recurrent neural network for predicting clusters are similar to that for exact positions, except that the dimensions of all parameters are changed to fit the inputs and outputs. Another change is that the cross entropy loss is used as the loss function as we have a classification task.

In terms of hyper-parameters, a higher initial learning rate of 0.005 would ensure convergence. Since the training is observed to be much faster, different numbers of hidden units are tuned for all the tasks and are discussed in the results section.

VII. Results and Analysis

A. *Predicting exact location*

Recall that all models predicting exact locations of birds used R^2 error as a scoring function to evaluate their performance. From the definition of R^2 score presented above, the maximum possible value for R^2 score is 1, and a negative value for R^2 score indicates that the model performs worse than predicting the mean values for every sample.

1) Linear models:

a) Linear regression: The following table describes the results obtained for Linear Regression:

Grouping by	Training R^2 Error	Testing R^2 error	Average distance to real positions (km)
4 days	0.927	0.943	80
7 days	0.868	0.904	114
30 days	0.631	0.753	190

Table II: Errors in predicting exact positions with linear regression

The above table suggests that the linear regression model by itself performs quite well according to the R^2 score, which has a maximum of 1.

b) Lasso regression: The following tables describe the results obtained for Lasso regression with $\alpha = 0.1$ and $\alpha = 0.00001$:

Grouping by	Training R^2 error	Testing R^2 error	Average distance to real positions (km)
$\alpha = 0.1$			
4 days	-0.002	-0.001	466
7 days	-0.007	-0.001	460
30 days	-0.016	-0.028	428
$\alpha = 0.00001$			
4 days	0.933	0.899	80
7 days	0.866	0.916	113
30 days	0.672	0.665	201

Table III: Errors in predicting exact positions with Lasso regression and $\alpha = 0.1$ and $\alpha = 0.00001$

From this result, we can conclude that using Lasso regularization does not improve performance of the model by much. Only when the value of α is small (0.00001), the Lasso regularized model performs as well as the linear model, which is the case theoretically since setting $\alpha = 0$ for Lasso regularization is equivalent to using a pure linear regression model.

c) Ridge regression: The following tables describe the results obtained for Ridge regression with $\alpha \in \{0.1, 0.01\}$, $tolerance\_threshold = 0.00001$ and no limit on $iteration\_count$:

Grouping by	Training R^2 Error	Testing R^2 error	Average distance to real positions (km)
$\alpha = 0.1$			
4 days	0.925	0.955	74
7 days	0.872	0.872	126
30 days	0.271	0.734	195
$\alpha = 0.01$			
4 days	0.929	0.928	79
7 days	0.871	0.864	127
30 days	0.665	0.733	191

Table IV: errors in predicting exact positions with Ridge regression and $\alpha \in \{0.1, 0.01\}$

Ridge regression results suggest that it performs slightly better than pure linear regression with a relatively large $\alpha = 0.1$, and becomes comparable to the pure linear regression model when α is small (which agrees with the theoretical result that having $\alpha = 0$ is equivalent of using a pure linear regression model).

d) Feature weightings for linear models: One of the advantages of linear regressions is that their results are very interpretable to extract the significance of the features used. This can promote better models in the future, as well as aid predictions as the climate changes. The following figures describe the most relevant features extracted from a Lasso regression with $\alpha = 0.00001$.

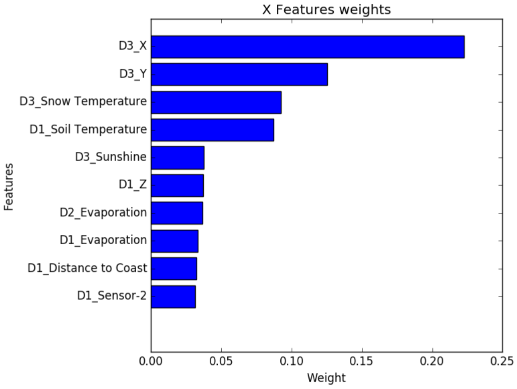

Figure 4: Top 10 relevant features in predicting exact location for X

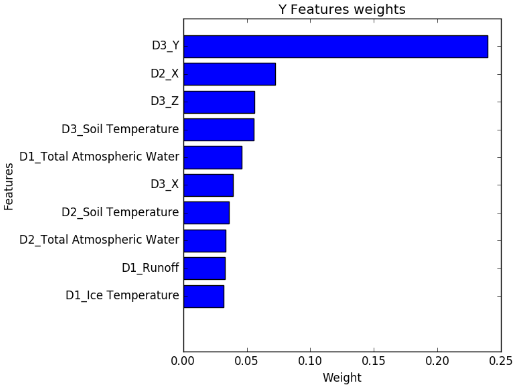

Figure 5: Top 10 relevant features in predicting exact location for Y

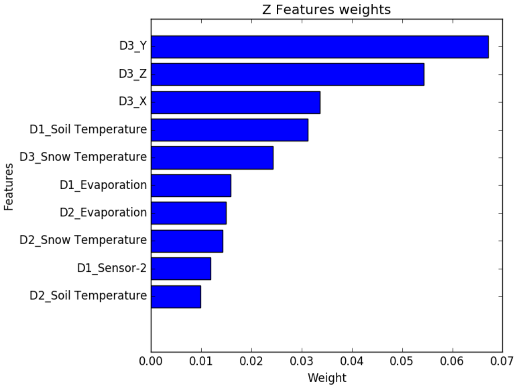

Figure 6: Top 10 relevant features in predicting exact location for Z

Figures 4, 5, and 6 show that birds are primarily classified based on their previous position. D1, D2, D3 represent 1, 2, and 3 days before, respectively. The next most significant features that are consistently represented for X, Y, and Z predictions are soil and snow temperature. Notably, month of the year is also a feature but is not in the top 10 features for any of the three outputs.

2) RNN: Similar to the linear models, 90% of the examples are used for training, the model is tested on the rest of the data set. The results are as follows:

Grouping by	Training R^2 error	Testing R^2 error	Average distance to real positions (km)
4 days	0.861	0.678	238
7 days	0.833	0.669	326
30 days	0.839	0.494	553

Table V: errors in predicting exact positions with RNN

3) Visualizing the result: To better visualize the performance of the models in predicting the exact locations of the birds, we plot the actual positions and their associated errors of the birds for the Lasso regression analyzed in section VII-A1d above together with RNN for the time period of 30 days (group by month). The figures below describe the results of these two models on the map, in which we do find RNN to perform worse than Lasso regression, but the errors from Lasso is more varied compared to that of RNN (i.e. a large range of color on Lasso regression map).

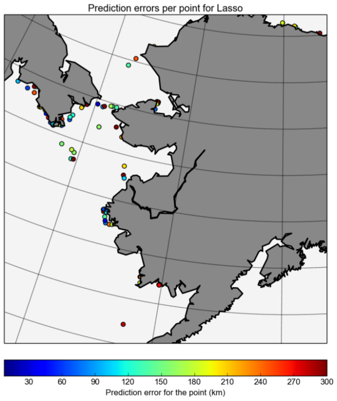
Prediction errors per point for Lasso
30
60
90
120
150
180
210
240
270
300
Prediction error for the point (km)

Figure 7: Visualization of error per point for Lasso regression

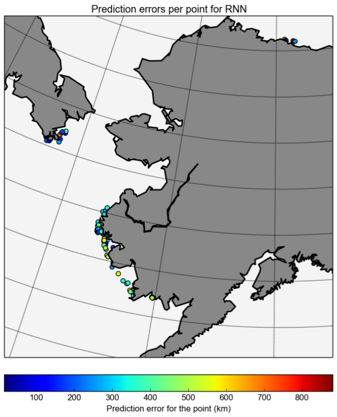
Prediction errors per point for RNN
100
200
300
400
500
600
700
800
Prediction error for the point (km)

Figure 8: Visualization of error per point for RNN

B. *Predicting by clusters*

Before exploring results for the models, we have to discuss the results obtained from K-means clustering algorithm. The below figure describes the percentage of error reduction where error is calculated as the sum of the distance from each point to its cluster's mean.

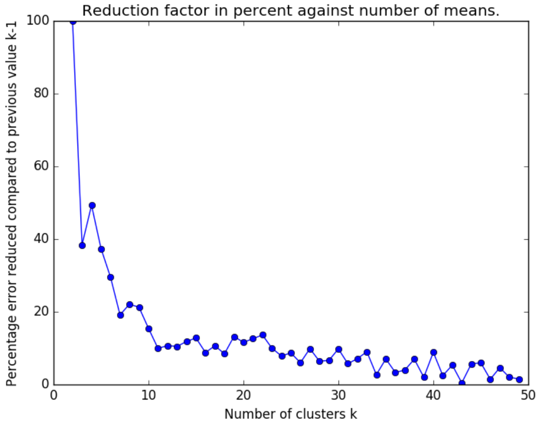

Figure 9: Reduction factor against number of means

The following figures visualize the results for K-means algorithm for $k = 11$ and $k = 20$:

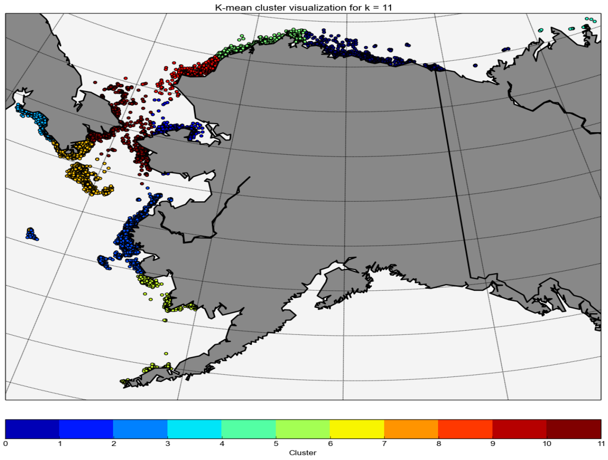

Figure 10: Cluster visualization for $k = 11$ means

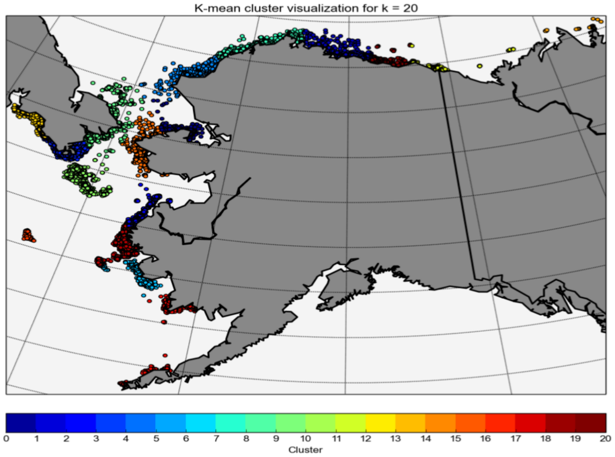

Figure 11: Cluster visualization for $k = 20$ means

Figure 9 reveals a sharp decrease in error with a k between 10 and 20. A lower k seems ideal, since there is only a minor reduction after $k = 10$. Further visualization in Figure 10 reveals that $k = 11$ is ideal. In contrast, $k = 20$ shown in Figure 11 includes clusters that divide populations on the same peninsula with no observable geographic distinction.

Similar to prediction of exact location of the birds, we also leave out 10% of the data as a test set.

1) Logistic Regression: The following table describes the result for Logistic Regression with different regularization strengths.

Grouping by	Training accuracy	Testing accuracy
$C = 1.0$		
4 days	0.795	0.838
7 days	0.795	0.813
30 days	0.601	0.595
$C = 0.5$		
4 days	0.782	0.838
7 days	0.782	0.811
30 days	0.592	0.595
$C = 0.1$		
4 days	0.737	0.785
7 days	0.737	0.766
30 days	0.567	0.595

Table VI: Accuracy in predicting clusters with Logistic Regression and $C \in \{1.0, 0.5, 0.1\}$

2) Random forest: A standard implementation of Random Forest from scikit-learn [9] is used. We try out with number of trees $m \in \{50, 100\}$. The following table describes the result for Random Forest with different number of trees.

Grouping by	Training accuracy	Testing accuracy
m = 50		
4 days	0.795	0.838
7 days	0.795	0.813
30 days	0.601	0.595
m = 100		
4 days	0.782	0.838
7 days	0.782	0.811
30 days	0.592	0.595

Table VII: Accuracy in predicting clusters with Random Forest and $m \in \{50, 100\}$

3) RNN: The results for RNN are as follows:

Grouping by	Number of hidden units	Training accuracy	Testing accuracy
4 days	25	0.761	0.829
	50	0.682	0.842
	100	0.612	0.861
	200	0.630	0.865
7 days	25	0.913	0.858
	50	0.880	0.897
	100	0.829	0.908
	200	0.830	0.906
30 days	25	0.795	0.842
	50	0.742	0.851
	100	0.759	0.812
	200	0.791	0.843

Table VIII: accuracy for classification with RNN

As can be seen from Table VIII, a more complex recurrent neural network tends to perform well when examples are grouped by shorter time period. For the grouping by 4 days data, 200 hidden units gives the best result. When the examples are grouped by 30 days, the number of hidden units needed drops to 50.

VIII. Discussion

The recurrent neural network does not preform as well as other models on regression. This is most likely due to an insufficient amount of data to learn complex temporal dependencies. The validation process for choosing the number of hidden nodes could also be improved. Although it is indicated that a model with few number of units learns fast in the beginning, this does not mean that such a model will have the lowest error after training. Thus the conclusion cannot be made that the recurrent neural network needs only 25 hidden units. The amount of data may still be insufficient.

The linear model outperformes the recurrent neural network by simplifying the predictions into comprehensible features. The linear model also reveals that after previous position, soil and snow temperature are the most powerful predictors of the next location. These are likely predictors of the optimal time for seasonal migration and far outrank month of year. This implies that common eiders move based on temperature, not the time of year, and are likely to adjust their migration time as the climate continues to change.

Meanwhile, the recurrent neural network perferomes the best of all models on classification. This could be due to a sweet spot in the amount of data needed to fully describe the movement of the eider. Classification is somewhat less precise than regression and as such might require less data to fully comprehend. A recurrent neural network with more hidden units works better with a larger data set. When the examples are grouped by a longer time period, more examples are grouped to create new data points. Thus a smaller data set is presented. While the number of hidden units needed drops from 200 to 50 (grouping by 4 days and 30 days respectively). This further indicates that a sweet spot in the amount of data needed is reached. The random forest model performs better than the linear model, but worse than the recurrent neural network. Random forests are better able to capture highly nonlinear data, whereas the linear model is constrained to more simplistic interpretations of the input space.

The greatest limitation is that the dataset we get is not large enough. With 110 individual common eiders, only 11 time series are used for testing after a 10% testing and training split. The testing error would be biased. One way to improve the results for a future would be to collect more examples. More experiments can be performed to find out the best combination of hyperparameters of the recurrent neural network. A long short-term memory (LSTM) could also be implemented.

Overall we are able to predict where would an common eider goes in the next few days and the next month with a error of 74 kilometers and 190 kilometers repectively. For the classification task, the next cluster a common eider will be in can be predicted with 91% accuracy. With an error of 74 kilometers, it is reasonable to set an area of that dimension to limit human activities such as fishing. With our models, humans activity can be better controlled to reduce interfere with the habitats of such graceful birds.

IX. Other attempts

An ant-colony clustering algorithm is implemented from scratch based on [4]. Ideally it would both preprocess the data for output class clusters in the classification task, as well as generate predictions for classification in a nearest-neighbor fashion. However, slow run times and redundant clusters render the model ineffective.

X. Statement of Contributions

HP implemented the first half and the last step of the data pipeline, as well as the linear models (Linear regression with Lasso and Ridge regularization) and the data visualization.

Yue implemented and tuned the recurrent neural network for both predicting the exact locations as well as classification task.

Corbin worked on data preprocessing in the data pipeline and feature interpretation, as well as the attempted ant-colony cluster implementation.

We hereby state that all the work presented in this report is that of the authors.

References

[1] Tucker Balch Adam Feldman. “Automatic Identification of Bee Movement”. In: *2nd International Workshop on the Mathematics and Algorithms of Social Insects* (2003). URL: https://smartsite.ucdavis.edu/access/content/user/00003578/Publications/Joshi_Schank2003.pdf#page=55.

[2] Galbraith C.A. Stroud D.A. (eds). Boere G.C. “Waterbirds around the world:A global overview of the conservation, management and research of the world’s waterbird flyways”. In: *Stationery Office* (2006). URL: http://jncc.defra.gov.uk/PDF/pub07_waterbirds_part6.1.6.pdf.

[3] H.J.W.M Cremers Theunis Piersma C.J Camphuysen C.M Berrevoets. “Mass Mortality of Common Eiders (Somateria mollissima) in the Dutch Wadden Sea, Winter 1999/2000: Starvation in a Commercially Exploited Wetland of International Importance”. In: *Biological Conservation* 106.3 (2002), pp. 303–317. URL: https://www.researchgate.net/publication/223837283_Mass_Mortality_of_Common_Eiders_Somateria_mollissima_in_the_Dutch_Wadden_Sea_Winter_19992000_Starvation_in_a_Commercially_Exploited_Wetland_of_International_Importance.

[4] Baldo Faieta Erik Lumer. “Diversity and adaptation in populations of clustering ants”. In: (1994). URL: http://www.cs.sun.ac.za/rw711/tuts/lumer94.pdf.

[5] T. Guilford et al. “Migration and Stopover in a Small Pelagic Seabird, the Manx Shearwater Puffinus puffinus: Insights from Machine Learning”. In: *Proceedings: Biological Sciences* 276.1660 (2009), pp. 1215–1223. ISSN: 09628452. URL: http://www.jstor.org/stable/30243982.

[6] PhD Kristin M. Branson. “Automatic Tracking and Behavior Analysis of Model Organisms”. In: *Howard Hughes Medical Institute* (2010). URL: http://www.hhmi.org/research/automatic-tracking-and-behavior-analysis-model-organisms.

[7] Laura McKinnon, H. Grant Gilchrist, and Kim T. Scribner. “Genetic evidence for kin-based female social structure in common eiders (Somateria mollissima)”. In: *Behavioral Ecology* 17.4 (2006), pp. 614–621. DOI: 10.1093/beheco/ark002. eprint: http://beheco.oxfordjournals.org/content/17/4/614.full.pdf+html. URL: http://beheco.oxfordjournals.org/content/17/4/614.abstract.

[8] James Meikle. “Tracking equipment could be harming wild birds, experts warn”. In: *The Guardian* (2011). URL: https://www.theguardian.com/environment/2011/sep/20/tracking-equipment-birds-health.

[9] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[10] MR Petersen and JM Schmutz. *Migration strategies of long-tailed ducks and common eiders during spring and fall.*

[11] Kristin Branson S.E. Roian Egnor. “Computational Analysis of Behavior”. In: *Annual Review of Neuroscience* 39 (2016), pp. 217–236. URL: http://www.annualreviews.org/doi/pdf/10.1146/annurev-neuro-070815-013845.

[12] M. Wikelski and R. Kays. *Movebank: archive, analysis and sharing of animal movement data.* 2016. URL: http://www.movebank.org (visited on 12/06/2016).