[redacted]

Algorithm:

We chose a rather unique approach to solving this clustering problem, using an unsupervised strategy (without taking into account training data) similar to agglomerative hierarchical clustering with single linkage. Our algorithm works by iterating through each observation in chronological order, where it predicts the ship that each observation is most likely a part of (or if it seems like a different ship altogether) based on the most recent observation (before it) of all ships already classified. We used this framework for both the situations where the correct number of ships is given and when not, adding an extra step to the end when the number is given.

In order to determine the ship of which a new observation is most likely a part, it uses each of these most recent observations to predict where the ship corresponding to them will be (in terms of latitude and longitude) at the time at which the new observation occurs, assuming they have been traveling at a constant angle and speed from the time of that observation. The prediction spatially closest to the new observation (in terms of squared Euclidean distance) is deemed to be the ship from which the observation occurred, unless the squared distance to the closest prediction is greater than 10^{-4} (or no other ships have been previously classified), at which point the new observation is set to be the first observation of a new ship.

If the predicted number of ships is greater than a given correct number of ships (with the difference being d), the algorithm takes remedial measures to decrease the number of clusters (representing the ship paths). It will keep reassigning the observations in the smallest d predicted clusters to the other clusters based on which one's most recent observation before the observation to be reassigned is closest to it, still in terms of squared Euclidean distance.

Exploratory Data Analysis:

We approached this problem from a more theoretical than observational perspective and performed data analysis more to test the assumptions of our theorized algorithm than to create it. Based on the description of the problem, we figured that the paths of the vessels would be arranged in a one-dimensional fashion such that the points on a path would form a figure similar to a line. Additionally, we figured that the paths should not intersect or come too close to intersecting at the same point in time because that would pose risk for collision. Given these factors, we reasoned that tracing the paths through time and predicting how their trajectory continues based on the most recent observation would be a reasonably effective method at classifying them assuming they could be accurately initialized.

We first analyzed the nature of the time differences between consecutive observations from the same vessel to test how reasonable it would be to assume that the speed and angle of motion for a vessel remains constant. As shown in Figure 1, most observations from a single vessel seem to occur approximately every 10 seconds, during which there would not reasonably be a significant change in trajectory. Sometimes a ship goes over a minute without transmitting data, but this does not occur often enough that it should significantly impact our predictions as a whole.

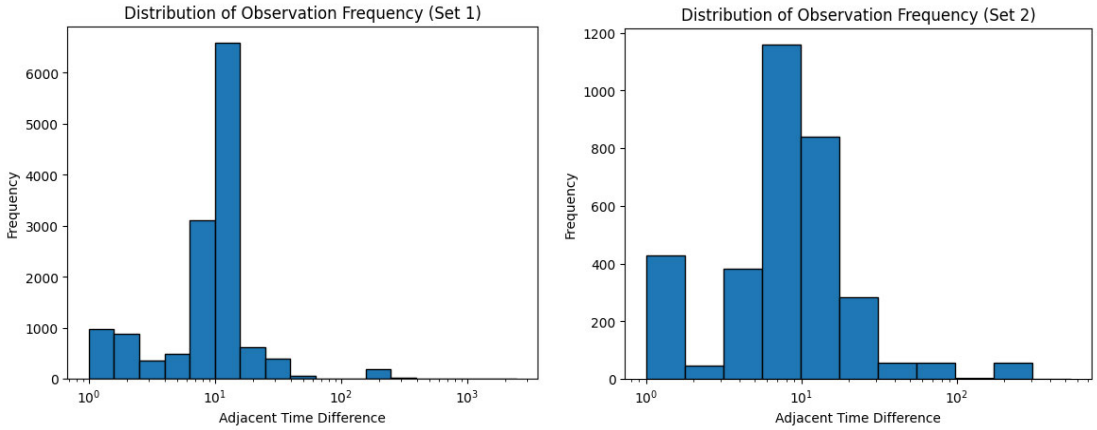

Figure 1: Consecutive Observation Frequency Histograms

Next, we analyzed our prediction function for determining where a ship should end up after a certain amount of time under the constant speed and angle assumptions. The histograms in Figure 2 showing the error in individual latitude and longitude predictions tend to be centered around 0, indicating that there is no heavy directional bias in the individual components of our estimation function. The scatter plots indicate that the vast majority of errors in terms of squared Euclidean distance are under 10^{-4}. While there does appear to be an increasing trend in error with time difference, we found that the threshold of 10^{-4} is enough to distinguish new vessel tracks while not creating new ones where there should not be.

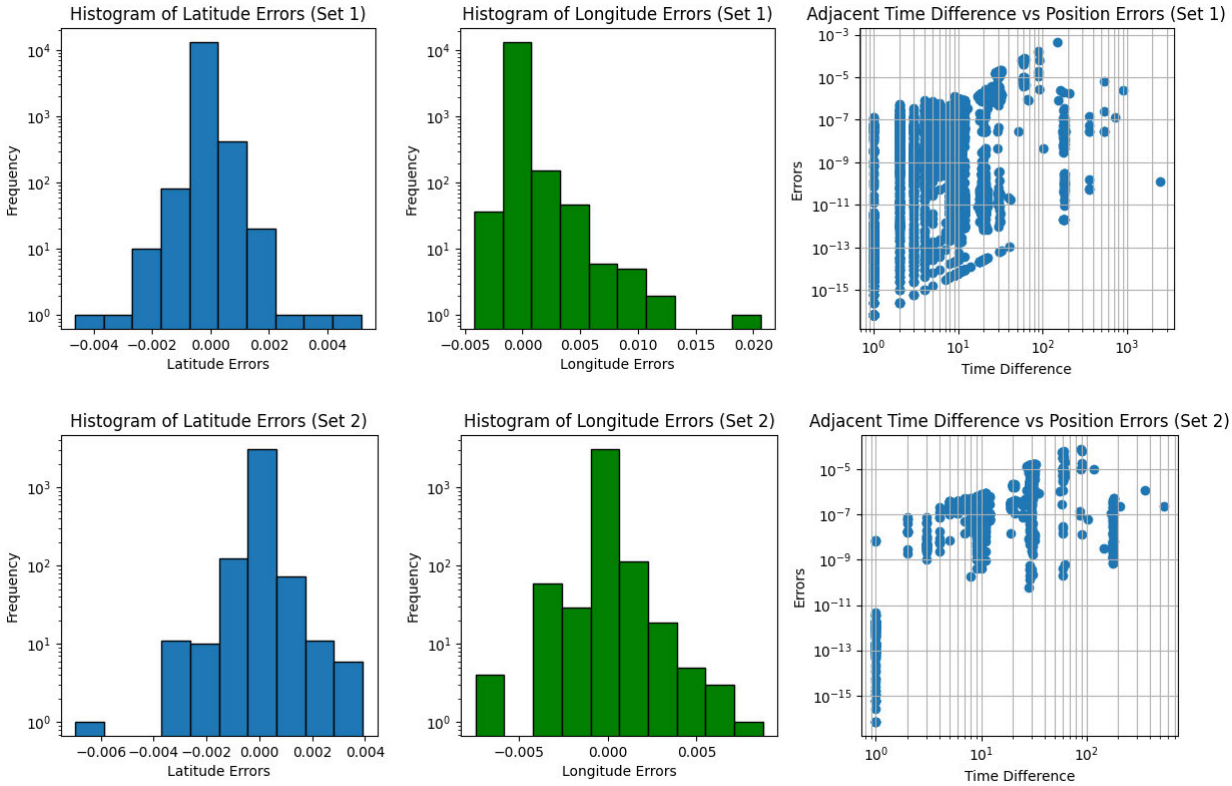

Figure 2: Position Prediction Error Plots

We did not perform any preprocessing on the data because we found that it would not be appropriate. There are no missing feature values in the data sets, so imputation is not necessary. Additionally, there were no speed values that reached the saturation limit of 102.2 knots, indicating that it would not be worth handling cases where this could happen. We did not perform any feature scaling because we catered to each feature's physical meaning individually rather than intercomparing them.

Testing:

Considering the nature of the data, we decided that supervised learning algorithms would not be very effective at solving the problem as the relationship a supervised algorithm might find between a class and its features may not extrapolate to new classes, in addition to not knowing the number of classes in a data set. As a result, we chose to consider unsupervised learning methods, specifically clustering.

To begin, we determined it was unfair to assume that observations in the same class would be close together in higher dimensional space. The speed, angle, location, and timestamp for observations are not necessarily similar to most other points in the cluster, so we determined that k-means clustering would be inappropriate, as the shape of the clusters would not have equal variance or equal size. Spectral clustering appeared to be an ineffective approach as well because it must know the number of clusters and is proven to be much slower when the number of samples is larger than the number of features. On the other hand, agglomerative clustering appeared to be an effective approach given our goal was to classify observations from the bottom up, examining observations chronologically and assigning them to cluster with the minimum dissimilarity. This process is what led us to develop our current algorithm,

To improve our algorithm, we implemented it in two ways and compared them. First, we considered the data chronologically, taking the most recent observation and attempting to predict where the ship should be at the time of the current observation. In our second implementation, we considered points reverse chronologically, taking the most recent observation and attempting to determine where it should have been considering the current observation.

One implication we were looking to evaluate was whether predicting class membership based on only one observation per prediction would indicate a propensity of our algorithm to make misclassifications cascade down paths enough to significantly affect our classification accuracy. By implementing it forwards and backwards we sought to see if there would be a significant discrepancy given the two directions for a given data set, which could lead us to explore combining both directions in a single run of the algorithm more effectively.

Direction	**Predicted K**	**Adjusted Rand Index**
Data Set 1		
Forwards	20	0.407
Backwards	21	0.350

Data Set 2		
Forwards	8	1.00
Backwards	8	1.00

Figure 3: Directional Differences in Adjusted Rand Index

Our backwards implementation performed a little worse than forwards for the first data set, as shown in Figure 3, but we determined that the difference of 0.057 for the ARI was not too significant and therefore did not indicate that the mistakes cascaded to a high degree.

In an attempt to improve our algorithm, we implemented a penalty that took into account the difference between the speed and angle of the current observation and each most recent observation from its potential classes within the threshold of the current observation. To do so, we expressed the speed and angle as a two-dimensional velocity vector for each observation and defined the error as the magnitude squared of the difference of the two vectors. We aimed to better differentiate between classes by assuming observations would be more similar in speed and angle to observations within the class than without, but we found that the error did not improve our results and chose not to implement it in our final solution.

To further assess the efficacy of our algorithm, we visualized and assessed the structure of the data in the third data set with respect to latitude, longitude, and time on a three dimensional plot, representing the position of a vessel in space with respect to time. The reason we chose the third data set for analysis is because it is the set that will be tested on.

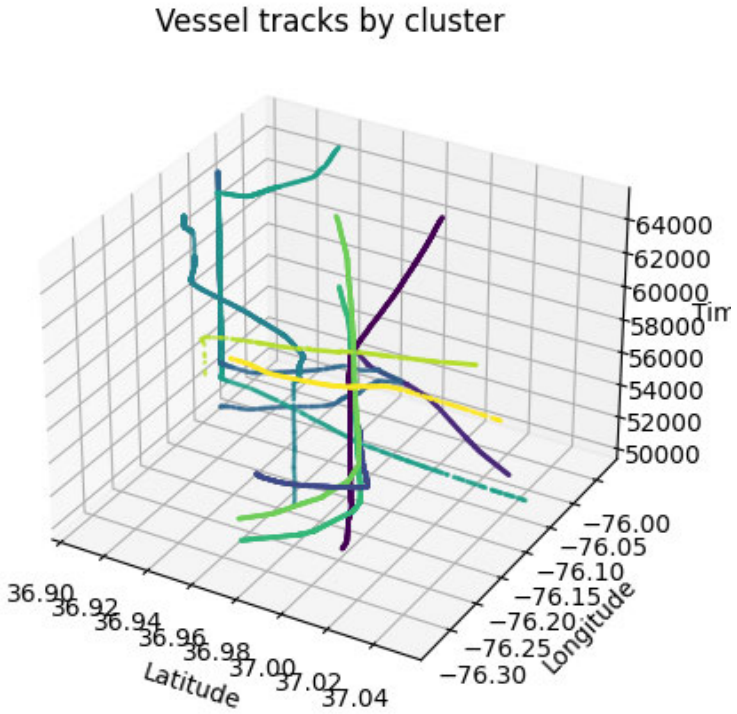

Figure 4: 3D Plot of Clusters on Data Set 3

Visually, the three-dimensional plot of our predictions appears to be sound based on the assumptions we made on the structure of the data. Each cluster on the plot follows a one-dimensional pattern, forming lines in three-dimensional space instead of forming geometric shapes. Upon further inspection, we found that there are no intersecting lines or very close paths when the vessel locations with respect to time as well. The scale of the graph is very large making it appear as if the lines intersect, but a close look at the plot shows that the seemingly overlapping lines are easily distinguishable from each other. Our algorithm corroborates this by classifying the points as different vessels as well, shown below.

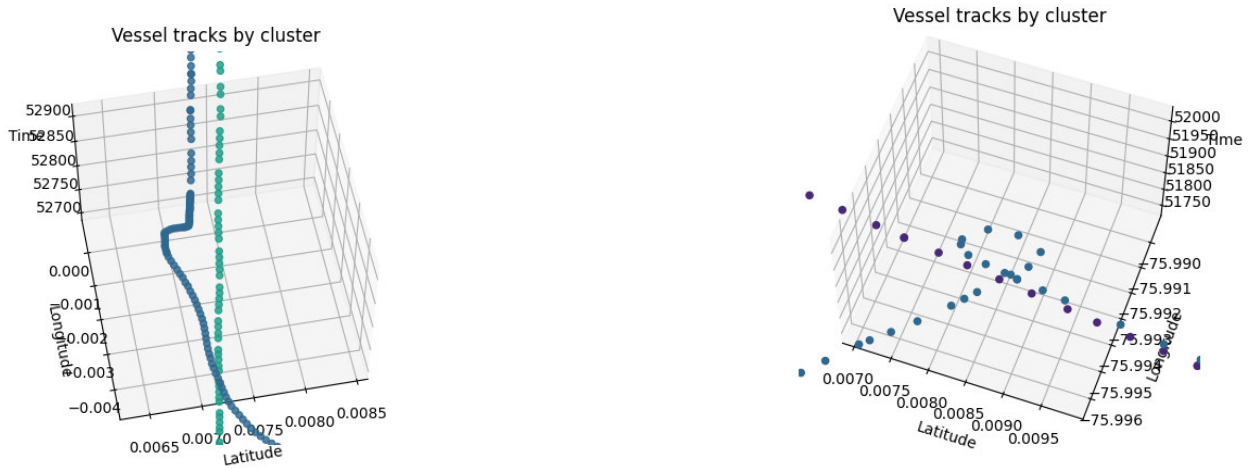

Figure 5: Close-up on 3D Plot

Efficacy

If implemented as a multiple target tracking system, our algorithm will work best with tracking vessels that transmit data frequently, are far apart from each other, and take a while to change speed and direction. Our algorithm will deal with gaps in data by assuming that the vessels will continue the trajectory from its most recent observation, though this assumption becomes more likely to be false the larger the gap. In the worst case, our algorithm will classify vessels that start transmitting data again after a large gap as new vessels, though this is less serious than incorrectly classifying them as another vessel that is not new.

In a real world setting, the number of vessels, the number of observations for a given vessel, and the shape of the distribution may vary wildly, so ARI, which makes no assumptions about cluster structure, should be a sufficient method to measure model accuracy. Utilizing ARI lets us determine how well our system can distinguish a vessel from other vessels by focusing on the agreement between pairs within and between clusters , additionally letting us observe how much better our algorithm performs when compared to chance. Other than requiring labeled data to evaluate the system, ARI does have the drawback of being susceptible to cluster size, putting less importance on correctly classifying observations in smaller clusters. If we were to use ARI as a metric on accuracy, we would be prioritizing correctly classifying vessels with more transmissions rather than equally considering transmissions from all vessels. An ARI above 0.9 should reflect a very good overall agreement of vessel classification between pairs.