ECM3401:Individual Literature Review and Project

Detection of Anomalous Vessel Behaviour between Wales and Northern Ireland

Understanding journeys between Wales and Northern Ireland

Jack Shaw

Abstract

This paper presents the development of tools to identify anomalous ship behaviour, which is crucial for enhancing Maritime Domain Awareness (MDA). Focusing on journeys between Wales and Northern Ireland, this paper leverages Automatic Identification System (AIS) data to discern normal vessel behaviour from anomalies. A range of anomalies are detected, including anomalous ship movements and positions, ships with invalid identification numbers, and incorrectly reported destinations. Anomalous ship movements and positions are detected using a Bi-directional Gated Recurrent Unit (BiGRU) Recursive Neural Network (RNN) to create a normal model. Deviations between this normal model and actual positions are classified as anomalous. Previous papers only detected these anomalies in large cargo and passenger ships. However, this paper uses a model for both small and large ships to detect anomalies in all ships.

I certify that all material in this dissertation which is not my own work has been identified.

Signed: Jack Shaw

May 2024

Contents

1 Introduction

This paper documents the creation of a set of tools to identify if a ship's behaviour is anomalous. Anomalous ship behaviour is when ships are involved in illegal or dangerous activities (e.g., speeding, unsafe navigation) or an emergency (e.g., man overboard, collision). Identifying anomalous ship behaviour improves Maritime Domain Awareness (MDA), the degree to which a country understands the factors which effect its maritime security, safety, economy, or environment [1, 2], by reducing the number of ships that need to be investigated. MDA is difficult and expensive to maintain due to the immense number of vessels that travel through the UK's waters (with 86,600 cargo vessels entering the UK's ports in 2022) [3] and the large size of its marine area, which is over three times larger than its land area [4]. Improving MDA is especially important for island nations like the United Kingdom, whose maritime economy comprises an estimated 8.1% of gross value added [5]. Further, security risks at sea documented by the Maritime Risk Assessment, highlighting terrorism, cyber-attacks, drug smuggling, and human trafficking [6]. The creation of a cheap to operate, easy to use set of tools for the improvement of MDA an important step towards addressing the concerns raised by the Maritime Risk Assessment.

This paper focuses on journeys between Wales and Northern Ireland as the shipping routes have been particularly affected by the Northern Ireland Protocol [7] and the following unstable political climate, so there are more potential threats. Further, the region has extensive station coverage, which guarantees abundant high-quality vessel data and is a high-traffic area [8].

An anomaly is a deviation from normal vessel behaviour [9]. Normal vessel behaviour is the behaviour the majority of ships follow. A ubiquitous reporting system that provides frequent updates and additional information, such as ship characteristics, is necessary to model normal vessel behaviour effectively. Automatic Identification System (AIS) fulfils these requirements and has been used to detect anomalies [10, 11]. AIS is a self-reporting tracking and monitoring system that helps to identify and locate vessels by exchanging electronic data (location, course and speed) with nearby vessels, ground-based stations, and satellites. Despite its advantages, AIS has issues with data quality, such as missing messages and human error leading to inaccurate information [12]. These issues are mitigated by repairing the missing data and identifying inaccurate information.

There are two shipborne AIS transponder classes, A and B. Class A is mandated for ships over 300 gross tonnage engaged on international voyages, 500 gross tonnage not engaged in international voyages, and all passenger vessels [13]. Class B is used by other vessels, including domestic commercial vessels and pleasure crafts. The main difference is that class B transmits less frequently, provides less information, and is not required.

Shipborne AIS transmits different types of messages. The most crucial information for MDA is the positional messages relating to a ship's current position (e.g., coordinate position, course, speed) [13]. Furthermore, static messages pertain to the ship's static characteristics (e.g., name, type and dimension) and voyage-related information related to the ship's journey (e.g., destination and estimated arrival time). A human inputs the static and voyage-related data with messages sent every six minutes. The transmission interval of positional messages depends on the current speed and status of the ship. A ship's position, speed and course are detected using different sensors on the ship [14].

The transponders which report positional messages use different definitions for speed, course, and heading. Speed Over Ground (SOG) is the vessel's speed relative to a surface of the earth using knots as the units [13]. Similarly, Course Over Ground (COG) is the vessel's direction of movement, between two points, with respect to the surface of the earth using degrees as the units. COG and SOG are used to take drift into account. The heading is the direction of the bow, using degrees. The Maritime Mobile Service Identity (MMSI) is a unique nine-digit identifier provided to an AIS transponder (e.g., a vessel or station). It is used to identify all ships across each dataset [15].

2 Literature Review

Best practice vessel behaviour includes entering correct information into the AIS, having your AIS active, following typical routes (e.g., shipping lanes and fishing grounds), and ensuring safe speed. Most ships follow these to ensure reasonable safety, transparency, and reputation; this behaviour is the norm. An example of normal behaviour is shown in appendix A, a vessel travelling at safe speeds, with its AIS active and along its ferry route. The anomalous behaviour of a ship can be summarised as actions that deviate from the normal [11]. As discussed in the review paper by [9], the anomalous behaviour of vessels can be subdivided into five types: positional (vessels deviating from a usual route or position; appendix B), kinematic (vessels deviating from a normal speed or course; appendix B), contextual (vessels whose deviating behaviour is related to a context such as port activity or weather conditions), data-related (involving missing, incomplete or irrational data; appendix C) and complex, whose specific behaviours are captured by detecting multiple anomalies. This paper aims to identify different positional, kinematic, and data-related anomalies.

2.1 Positional and Kinematic Anomalies

Positional and kinematic anomalies are detected together since a ship's position influences its speed and course, and vice versa [11]. For example, a ship manoeuvres more cautiously in a river than in the deep sea. At the recommendation of the literature, three steps are used to detect positional and kinematic anomalies. First, a normal model is created. Second, the normal model predicts the ship's likely position. Then, deviations between the predictions and the actual data are calculated; when deviations reach a threshold, they are deemed anomalous [16].

2.1.1 Modelling Normal Vessel Movements

There are three approaches to modelling normal vessel movements: data-driven, signature-based, and a hybrid of both [9, 16]. Data-driven uses an unsupervised algorithm and historical data, while signature-based uses predefined patterns (for example, turns or holding patterns). Data-driven approaches require a large dataset of historical vessel activity to create an accurate model, which is now possible due to the expansion of ground-based and satellite AIS systems [17]. Due to the complexity and vast number of patterns, even with expert knowledge, defining what patterns to include is difficult. These limitations have resulted in the limited adoption of signature-based methods [9]. Therefore, a data-driven approach is being used.

For the algorithm to create an accurate normal model using a data-driven approach, it must be unsupervised and able to classify data as either normal or a data-related anomaly [9]. The algorithm must be unsupervised since supervised techniques require labelling, which is not feasible due to the size of the dataset and the difficulty of defining normal vessel behaviour. Density-Based Spatial Clustering of Applications with Noise (DBSCAN; [18]) meets these requirements since it is unsupervised, can classify data as part of a cluster or an outlier and can discover clusters of arbitrary shapes so it can cluster ship journeys.

DBSCAN is a density clustering algorithm created by [18]. Clustering is an unsupervised machine learning technique to identify and group similar data points in large datasets [19]. Specifically, DBSCAN clusters dense points separated by low-density regions. Any points within the low-density regions are noise points, a data-related anomaly (appendix C).

Despite DBSCAN's advantages, it requires determining input parameters [17]. DBSCAN results are sensitive to the values of these input parameters, so it is prudent to select them correctly. DBSCAN's minimum points parameter is the lowest required points inside each cluster otherwise the points are classified as outliers [11]. Based on the experiments in [18], the minimum points parameter can be set to the dimensions of the data multiplied by two. This is done to eliminate the minimum points parameter so that epsilon, the other parameter, could be found. DBSCAN's epsilon parameter represents the maximum distance between two points in the same cluster, the density of each cluster [11]. A sorted k-distance graph is used to find the epsilon value. The k-distance graph is the total distance of each point's k-nearest neighbours (KNN) sorted from largest to smallest. The k number of neighbours is set to the minimum point parameter. The range of epsilon values is found by identifying the elbow on the graph when the distance stops decreasing greatly but before it flattens out [18]. Since the elbow has a range of values, experiments are performed to find the best epsilon value in this range. The best epsilon value in these experiments has the highest mean silhouette score. Silhouette score is a metric for assessing how well-defined and distinct the clusters are [20].

2.1.2 Predicting Positions, Course and Speed

Vessel movements are often predicted using an Artificial Neural Network (ANN) [11, 17, 21], which maps an input to an output trained on example input-output pairs derived from existing data. This enables effective prediction of vessel movements by identifying complex non-linear relationships between input and output data.

ANNs require multiple datasets to train and validate their results. The neural network learns from the training set by adjusting its weights and biases to reduce the loss between predictions and the training set [22]. The testing dataset assesses the ANN performance on unseen data after the training is complete, the validation assesses during the ANN's training. As done in [11], the training and validation are split randomly from the same dataset. The testing dataset is collected separately to assess that the model has been generalised.

Recursive Neural Networks (RNN) are a class of ANNs [23]. They handle sequential data and can selectively pass information across sequences, allowing for sequences of elements dependent on each other. An RNN is particularly well suited for vessel journey data as it is sequential, and the previous position affects the next [24]. A more traditional class of ANN could have been used, such as a Feed-forward Neural Network. However, [17] demonstrated that an RNN had 12.8% less predictive error than the Feed-forward Neural Network on average when predicting vessel positions.

RNNs have been most successful at predicting vessel movements using Bidirectional Gated Recurrent Units (BiGRU) layers. A BiGRU contains a forward and backwards Gated Recurrent Unit (GRU) within each layer, improving the efficiency of weight updates and the speed it trains [11]. A GRU is a variant of Long Short-Term Memory (LSTM) that combines the forget gate and input gate in LSTM into an update gate, introduced by [25]. The update gate controls the flow of information within the network, addressing the vanishing gradient problem. The vanishing gradient problem is where the gradients of the loss function become extremely small during training, slowing or even preventing training [26]. The loss function is a method for evaluating the model's fit to the data. BiGRUs are preferred over LSTMs [17] as BiGRUs were shown to have a 96.35% accuracy and only a 1.67% false positive rate when detecting positional and kinematic anomalies in ships as supposed to an accuracy of 88.15% and a 9.79% false positive rate with LSTM [11].

The RNN is trained using the sliding window method, as it has been shown to train RNNs better than conventional methods [27]. The sliding window method uses multiple previous positions to teach the model the next position, sliding through each position until the journey is complete. The number of positions in the window depends on the window size selected. Since each sliding window can only train and predict one column, there are four different models to predict COG, SOG, latitude and longitude columns. Each model has parameters selected using experiments, selecting those with the fastest training time and lowest predicted error. Parameters include the loss function, optimisation function, batch size, window size and which columns to include in the sliding window. The model aims to minimise the loss function during the training process (difference between the actual and predicted values) [28]. Since this is a regression problem, choosing an appropriate loss function is necessary. The optimisation function's primary function is to adjust the network's weights and biases during training to minimise the loss function [28]. The optimisation function achieves this by updating the weights based on the gradients of the loss function concerning those weights. The batch size is the number of samples processed by the model before updating the model's weights [29].

There are eight total RNN models, as each column is trained on two different vessel sizes: small and large. The RNNs are trained differently since the ship's size alters how it moves [30, 31] and the depth it can sail through [32]. For example, small ships change quickly (accelerating, decelerating and turning quickly), while large ships change slower. The sizes are classified using the area of the ship, calculated as length times the beam. The boundary between small and large ships is the mean ship area. The vessel's positions are then predicted using the RNN that matches its size. Creating separate RNNs allows all sizes and types of vessels to train the neural networks. This expands upon research identifying anomalous ship behaviour as [11, 17] only used tankers and cargo ships as they have the most predictable movements. However, anomalous behaviour is not exclusive to these vessels, so including all vessels in the prediction is vital. Using multiple RNNs mitigates the problem because the larger and more predictable tankers and cargo vessels do not impact the results of other smaller vessels.

2.1.3 Assessing Deviation from the Normal Model

The deviation between the actual and predicted journeys is calculated once the trained RNN has predicted a journey using the testing dataset. The Haversine distance between the actual and predicted positions is calculated as the deviation. Haversine distance assumes that the Earth is a perfect sphere [33], making it inaccurate but quick to calculate. These inaccuracies should not be a problem since they become problematic when the distances between the coordinates are large, which is not the case. When the deviation in metres reaches a low or high threshold, the vessel will be flagged as anomalous. The low threshold is 100 metres, and 250 metres for the high threshold as calculated in [11]. The SOG and COG's deviation is determined by calculating the absolute error between the actual and predicted values. If the resulting deviation is plus or minus 3% the actual value, it is highlighted as anomalous [11].

2.2 Data-related anomalies

AIS is an open system conceived to allow as many vessels to use it as possible. However, this openness has led to a lack of accurate and complete data: incorrect or falsified data may be entered [34], or AIS messages can be entirely missing [35]. Detecting data-related anomalies in AIS data is crucial to ensure they can be removed to detect positional and kinematic anomalies accurately, and they provide additional information about the AIS data.

2.2.1 Missing and Delayed Vessel Positions

AIS data can be missing due to transmission problems [36]. There are two ways transmission problems can be caused, even when in the range of an AIS station. AIS signals are transmitted at a very high frequency. Due to its high frequency, it is prone to degradation due to environmental conditions or shadowing from landmasses. Second, AIS stations can become overwhelmed with messages, resulting in loss of messages [37]. Since the data collected is from stations, there is likely to be missing AIS data from beyond the range of AIS stations.

AIS positional data is transmitted at different intervals depending on the speed and if it changes direction, as shown in appendix D. This is considered when determining if the positions are missing or delayed data. Since this is the nominal reporting interval, not all Class A ships comply with these intervals, with [38] finding that 42.3% of positional messages had an interval longer than nominal. Delayed data is counted as less than double the nominal for the current condition but greater than the nominal reporting interval. Missing data (that has to be repaired) is counted as an interval double the nominal interval as the time gap ensures that at least one position can be repaired.

Cubic Hermite Spline Interpolation [39] is used to restore the missing data. Interpolation is a method to determine missing values from existing values and is popular in restoring values in time series, like AIS data [40]. Cubic Hermite Spline interpolation has been chosen since it has been shown to repair AIS data accurately when slightly to moderately degraded. [40] showed that Cubic Hermite Spline Interpolation outperformed other interpolation methods (Linear and Cubic Spline Interpolation), producing a lower error between the interpolation and actual values. It also produced a more realistic track, a safe distance from land. However, it only performs best when the data is not significantly degraded. In a different experiment on less degraded data, [10] found that Cubic Hermite Spline Interpolation performed slightly worse than Cubic Spline Interpolation, making it sufficient to repair the missing positional data.

2.2.2 Incorrect Journey Destination

The destination is entered as part of the voyage-related AIS data [13]. A human enters the destination, so it may not be correct and needs to be verified. It is verified using the positional data of the vessel's journey, comparing the places the ship moored with the reported destination.

Rules defined by [41] are used to create journeys. The rules are as follows: All ships are assumed to be sailing on journeys at the beginning of data collection, a ship begins mooring if its SOG is lower than half a knot for five consecutive hours, a ship stops mooring if its SOG is higher than one knot and any two consecutive moorings are a journey. When a ship begins mooring, a position is registered. The nearest port within a certain distance is counted as the destination port and then compared to the entered destination. Since there is only a certain collection area, any ships that leave the collection area are assumed to have completed a journey. The entered destination is unlikely to contain the exact port name, so the port name needs to be extracted from the destination, and each port needs an alternate name.

2.2.3 Invalid MMSI Numbers

Since MMSI numbers are used to identify ships and any message they transmit using the AIS, they must be valid. A MMSI number should be between nine and four digits in length and begin with a three-digit code, the Maritime Identification Digits (MID) [15]. The MID denotes the area that has jurisdiction over the ship.

2.3 Evaluation Strategy

One of the challenges in detecting positional and kinematic anomalies is a lack of ground truths to evaluate the effectiveness of the results [42]. This lack of ground truth partly stems from the difficulty in defining anomalies. Further, [42] show that a common problem when evaluating is using synthetic data, such as creating artificial journeys and adding anomalies to real-world journeys. Evaluating using synthetic data limits the ecological validity of the model.

A better way to evaluate models is to use labelled real-world journeys, which can act as a ground truth as used in [11]. This is done by labelling a journey's anomalies and comparing these to the model's predicted anomalies. The model classifying anomalies at the correct time, duration, and type indicates the effectiveness of the RNN. Another method to provide a ground truth is to compare vessel journeys in a storm with calm weather [43]. The storm should result in higher anomalous behaviour, with vessels avoiding the storm or the winds affecting the ship's movements.

Error metrics evaluate how the RNN predictions compare to the actual positional data, with lower errors indicating better predictions. Two error metrics evaluate the RNN models: Mean Absolute Error (MAE) and Mean Squared Error (MSE). MAE represents the average magnitude of errors between the predicted and actual values as shown in (01) [44]. MSE represents the average squared magnitude of errors between the predicted and actual values as shown in (02) [45]. For (01) and (02), the y_i symbol denotes the actual data for the ith observation and $\hat{f}(x_i)$ denotes the prediction for the ith observation. Since MSE is squared, it is more sensitive to outliers than MAE, so it is used to highlight any outliers. These error metrics are used to evaluate the model whilst training (with training and validation errors) and on unseen data (predictive error).

$$\text{MAE} = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} ||y_i - \hat{f}(x_i)|| \quad (01)$$

$$\text{MSE} = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2 \quad (02)$$

A similar approach is used when evaluating the destination validation algorithm. When a ship reports a destination and arrives at the specified destination, this is classified as a valid destination. If a ship does not arrive at its specified destination, this is classified as an invalid destination. If a ship has no destination, this is classified as no destination entered. If a ship's specified destination is outside the collection area or cannot be decoded due to a spelling mistake or a ship not inputting a location, then this is classified as unable to be matched to a destination. The accuracy and false positive rate are used to evaluate the success of the destination validation. The accuracy is calculated using the equation (03) [46]. True positives are any actual valid destinations that are predicted to be valid. In contrast, true negatives are invalid or missing destinations predicted to be invalid or missing. False negatives are any valid destinations that are predicted to be invalid or missing a destination. In contrast, false positives are invalid or missing destinations predicted to be valid. The inability to match a destination classification is not included since this is intended to filter out spelling mistakes and destinations beyond the collection area. The false positive rate is calculated using equation (04) [46].

$$\text{Accuracy} = \frac{\text{True positives+True Negatives}}{\text{True positives+True negatives+False positives+False negatives}} \qquad (03)$$

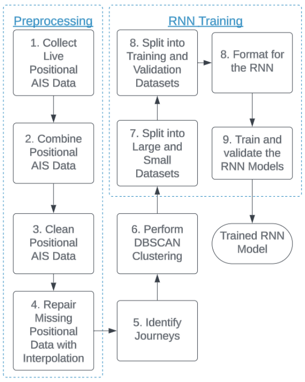
$$\text{False Positive Rate} = \frac{\text{False positives}}{\text{True negatives+False positives}} \qquad (04)$$

2.4 Hypotheses

Hypothesis one is that positional and kinematic anomalies will be correctly classified with an accuracy of 96.35% or greater [11]. Hypothesis two is that the positional and kinematic anomalies will have a false positive rate of 1.67% or lower [11]. Hypothesis three is that destination verification has an accuracy of 84.1% or greater [41]. Hypothesis four is that destination verification has a false positive rate of 2% based on [11]. Any higher false positive rate could lead to mistrust with the results provided, hence why 2% is used. Hypothesis five is that using DBSCAN improves the detection of positional and kinematic anomalies. Hypothesis six is that MIDs can be determined to be valid or invalid with an accuracy of 100%. Hypothesis seven is that MIDs can be determined to be valid or invalid with a false positive rate of 0%. Hypothesis six and seven were chosen with high accuracy and low false positive rates since it should be trivial as it only requires validating the MIDs to a reliable dataset.

3 Design and Implementation

The RNN is trained using the combined positional AIS dataset. Before the combined positional AIS dataset can be used for training, it must first be pre-processed, journeys identified, clustered and split into datasets for each RNN model (Figure 1). Once the models have been trained, the testing dataset follows a similar process, as shown in Figure 2. The trained model is used to predict the testing dataset with deviations in this dataset indicative of anomalies.


Preprocessing
RNN Training
1. Collect Live Positional AIS Data
2. Combine Positional AIS Data
3. Clean Positional AIS Data
4. Repair Missing Positional Data with Interpolation
5. Identify Journeys
6. Perform DBSCAN Clustering
7. Split into Large and Small Datasets
8. Split into Training and Validation Datasets
8. Format for the RNN
9. Train and validate the RNN Models
Trained RNN Model

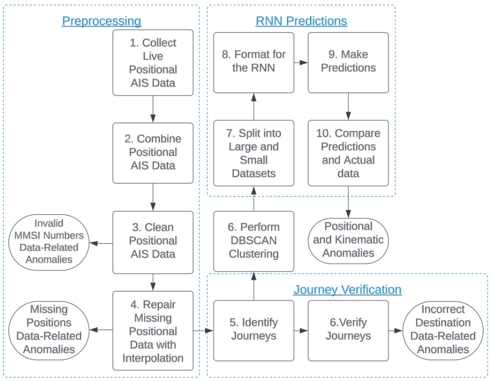

Preprocessing
RNN Predictions
1. Collect Live Positional AIS Data
2. Combine Positional AIS Data
3. Clean Positional AIS Data
4. Repair Missing Positional Data with Interpolation
Invalid MMSI Numbers Data-Related Anomalies
Missing Positions Data-Related Anomalies
8. Format for the RNN
9. Make Predictions
7. Split into Large and Small Datasets
10. Compare Predictions and Actual data
6. Perform DBSCAN Clustering
Positional and Kinematic Anomalies
Journey Verification
5. Identify Journeys
6.Verify Journeys
Incorrect Destination Data-Related Anomalies

Figure 1 (left): A flowchart to show how the combined positional dataset is processed and used to train the RNN model.

Figure 2 (right): A flowchart to show how the testing dataset is processed and where different anomalies are detected.

3.1 Data Pre-processing

Pre-processing is a crucial step before analysing the static and positional data. This ensures the data is reliable, accurate and correctly formatted for further analysis. Initial pre-processing is performed across the initial four steps (Figures 1 and 2): collection, combination, cleaning, and repairing missing data. Additional pre-processing is required for the RNN to ensure it is in the correct format. Python version 3.11.5 [47] is used on a local machine to perform the pre-processing. The local machine has an Intel i5-10300H CPU, 16 GB of memory and NVIDIA GeForce GTX 1650Ti GPU. The Python library Pandas 2.1.1 opens, saves and manipulates the Comma-separated values (CSV) files [48].

3.1.1 Collection and Combination

The AIS positional and static datasets are collected from the AIS-Stream web socket Application Programming Interface (API) [49]. It is used since it provides free and real-time AIS data. The collection area is a box with the longitudinal and latitudinal coordinates of (51, -2.3) and (55.5, -7.4) over the Irish Sea. A collection box is used to reduce the number of ships being collected (more ships require more storage and processing time) and only to collect the ship journeys between Wales and Northern Ireland. An API key is used to authenticate the web socket API, which is pulled for a registry key to ensure the security of the key.

Different types of messages are received for ships within the collection area [49]. Positional vessel data is collected from "PositionReport" and "ExtendedClassBPositionReport" messages. "PositionReport" messages are from Class A AIS transceivers, and "ExtendedClassBPositionReport" messages are from Class B AIS transceivers. Since Class B AIS transponders transmit less information, any missing information (e.g., navigational status or rate of turn) is set to the missing or default value from [13]. The static and voyage vessel data is collected from "ShipStaticData" and "StaticDataReport." Since the "StaticDataReport" only transmits some static data and no voyage data, these are set to the missing or default value from [13]. The AIS data is collected and sorted into Comma-Separated Values (CSV) files of positional vessel data and combined static and voyage data.

The combined (training and validation before being split) data is collected from one week in late November 2023 (04/11 - 09/11) and one week in late January 2024 (09/01 - 16/01). The data is collected at different week periods to allow the model to generalise better, with each section starting and ending late at night to ensure that most ships were in port, not separating journeys. The testing data consists of a week in early March 2024 (04/03 - 11/03) and an hour in late February 2024 (15:00 - 16:00 on 26/02). The hour was introduced to add some anomalies to the dataset. After combination, there were 1,413,520 rows in the combined dataset and 1,111,221 rows in the testing dataset, which should provide sufficient data to train, verify and test the model.

Port data, used to identify and verify journeys, is from the World Port Index (WPI) [50] for larger ports and supplemented from Open Street Maps (OSM) [51] for smaller ports and marinas. Both datasets were used since the WPI is more reliable, originating from the National Geospatial-Intelligence Agency, and contains more information at each port (e.g., size, type and facilities) but only contains larger ports. The OSM data contains smaller ports and marinas and is constantly updated, but it can be unreliable since it is a collaborative project where users contribute data. Combining them mitigates the flaws of both datasets by verifying the OSM data and expanding upon the WPI. The port dataset contains 487 ports across Ireland, Wales, England and Scotland.

3.1.2 Cleaning and Formatting the Positional Data

Positional AIS data contain a timestamp in seconds since the message is generated [13] and a string data-time generated when the message is received from the AIS web socket. The original received string date-time is converted into seconds since the Unix Epoch, with a precision of up to a microsecond. The time the message is generated is created using the received time, taking away the seconds since the message is generated. Formatting it this way reduces memory usage per date-time from twenty-nine to eight bytes and is easier to compare and manipulate. The time since the message is generated is the definitive time the AIS data is created unless there are too many invalid timestamps within the dataset. Invalid timestamps can be corrected using the mean timestamps for that vessel, but this is only so effective. If 5% of timestamps are missing, then the received time is used as the definitive time for the AIS message.

Then, the positional data is cleaned by setting any out-of-range value to the no information available value [13]. For example, SOG has a range between 0 - 102.2 knots and any values out of this range are set to 102.3. Only values used by the RNN for predictions (SOG, COG, latitude and longitude) are removed to be repaired by interpolation.

Then, each column is rounded to the precision specified by [13]. For example, COG has a precision of 1/10, so it will be rounded to one decimal point. With each column rounded down, it can be converted to a smaller datatype to reduce memory usage unless precision would result in a significant loss of information. For example, SOG and COG are converted to a four-byte float, which should be sufficient as they only have one decimal point. However, latitude and longitude are set to an eight-byte float, as any loss in precision would result in a loss of information.

Then, any duplicate MMSI numbers and the definitive timestamps are dropped. This is done since MMSI numbers and the definitive timestamps act as the composite primary keys, and duplicates prevent repairing missing data as interpolation requires the timestamps to increase continually.

The static dataset follows a similar process, with received timestamps converted to time since the Unix Epoch, values set out of range to the no information available value, rounding columns to the correct precision, and dropping duplicates.

There are many missing timestamps, with 139,368 (9.86%) in the combined and 74,055 (6.66%) in the testing datasets. Since over 5% of timestamps are missing, the received timestamp is used as the definitive time instead of generation time, as generation time is too inaccurate. Three hundred twenty-nine duplicate positions were removed from the combined dataset, and zero duplicate positions were removed from the testing dataset. This is a small minority of the dataset, so it does not impact its overall quality by removing them. However, there are a higher number of out-of-range values with 11.75% of headings, 11.29% of turning rates, and 7.14% of COG values set to the out-of-range value or missing in the combined dataset. Despite this SOG, Latitude and Longitude, essential for predicting positions, were rarely missing or out of range. This resulted in 100,982 rows being dropped from the combined dataset and 94,370 from the test dataset. This is a substantial number of rows to be dropped but is mitigated by interpolating these removed rows; for the complete tables of missing values, refer to appendix E and F.

3.1.3 Repairing Missing Positional Data

As detailed in section 2.2.1, missing positional data is repaired through Cubic Hermite spline interpolation. However, if a considerable time gap (exceeding eight hours) exists, Cubic Hermite spline interpolation can no longer effectively repair the data as it leads to the generation of unrealistic values (e.g., courses over 360 degrees and negative) and erroneous ship trajectories, such as ships appearing to traverse over land.

Before repairing the missing positional data, each position needs to be categorised. As outlined in section 2.2.1, nominal reporting interval is the time gap between position messages that changes based on the ship's dynamic conditions. Using the nominal reporting interval and the actual time gap between the previous message, a position can be categorised as normal, delayed, in need of repair, or unsuitable for repair. Normal means that there is less than or equal to the reporting interval. Delayed means that it is greater than the reporting interval but less than double the reporting interval. In need of repair means the reporting interval is greater than and equal to double the reporting interval but less than an eight-hour time gap. Cannot be repaired have a greater than or equal to an eight-hour time gap. Delayed and Normal data need not be repaired since the time gap is too short to add a point between them. Any excessively degraded vessel journeys with more than two positions that cannot be repaired are removed from the positional dataset. When a vessel is in a port, no interpolation is performed to reduce unnecessary additional positions added to the datasets. The interpolated positions are cleaned to ensure that no impossible values have been introduced after interpolating, as in section 3.1.2.

After the repair process, the positional data is classified as follows: 6.29% normal, 7.19% delayed, and 86.42% requiring repair for the combined dataset. A mere 0.09% could not be repaired due to a time gap exceeding eight hours for the combined positions; for further details, refer to appendix G. Many rows were removed due to multiple large time gaps, with 1,144,403 from the combined dataset and 615,945 from the test dataset. The repair process involved interpolating 2,866,556 rows for the combined dataset, resulting in a repaired dataset with 3,034,691 rows. Similarly, the test dataset has 5,795,649 rows interpolated, leading to a total of 6,196,555 rows. This interpolation significantly reduced the average time gap of the combined dataset from 1331.46 seconds to 73.55 seconds and the test dataset from 118.98 seconds to 7.69 seconds. These figures demonstrate the repair method's effectiveness, showcasing how it has transformed the dataset by reducing the time gaps between positions.

3.1.4 Journey Identification and Verification

As discussed in section 2.2.2, journeys are any positions of ships between a mooring or leaving or entering the collection area. A ship begins mooring if its SOG is below half a knot for five consecutive hours and stops mooring if its SOG is over one knot [41]. A journey contains the journey identification number (a unique ID given to each journey), the ship's MMSI number, the departure port ID, departure time since Unix Epoch, destination port ID, and destination time since Unix Epoch. Journeys are created by looping through each vessel and then through each vessel's position. For each loop, a new journey is created if there is a time gap of eight hours. If the SOG is below half a knot and the ship is not moored, the time and position are recorded, and the ship is moored. If the sog is greater than one and the ship is moored, it is no longer moored, and if it was moored for over five hours, it finds the closest port within five kilometres of the ship's original mooring position. If no port is found, it marks the destination port as minus one, and if it does, it uses the port ID from the port dataset. The departure ID is the same as the previous journey's destination ID if it is the same ship. The beginning and ending port IDs are denoted by minus two. Each position is given a journey ID to ensure each position is linked to the journey ID.

For the combined dataset, 805 journeys were identified, while 1,024 were identified for the test. Journeys in the combined dataset take nearly seven hours and contain an average of 3769.80 positions. In the test dataset, journeys took nine and a half hours and contained an average of 6051.32 positions.

Journeys are verified based on the actual destination and the human-inputted voyage data. The verification is performed by comparing the voyage AIS data to the actual positional AIS data to verify its correctness. The verification uses the cleaned static AIS data and the journey positional AIS data for the test data. The test dataset was only used in the final results, and journey verification is not used to train the RNN model. The static data is classified as having no destination, unable to match the destination to a port, and being valid or invalid. There are several reasons why the destination could not be matched to a port, including the port not being in the port dataset, an alternate unknown name being entered, or a misspelling. The port data has been modified to include the United Nations Code for Trade and Transport Locations (UN\LOCODE) to reduce the chance of the destination not being found, as it was found that 25.96% of the static dataset has a UN\LOCODE destination. This dataset originates from [52].

Verifying the destination is performed in several steps. First, any static data without records within the positional data is removed. Then, any destinations that have no destination (e.g., blank or only contain the "@" symbol used to denote an empty message) [13] are classified. Looping through every row that has not already been classified, it is checked that the destination matches a port name, alternative name, or its UN\LOCODE. If the destination does not match, it is classified as such. Then, the closest positional message to when the static message was sent is retrieved for the same vessel. If the ports match, then it is valid else, it is not a valid destination.

3.2 Clustering

Clustering the positional data is performed by the DBSCAN algorithm, as discussed in section 2.1.1. DBSCAN works in several steps to cluster points. First, it randomly selects a point that has not been visited and points within the Euclidean distance of epsilon from it. It then recursively finds all of the neighbours to those neighbours until there are no more, completing the cluster. There are three types of points: core, border and noise points. A core point has at least the number of minimum points within epsilon distance of them, while a border point has fewer than minimum points but is still considered in that cluster. A noise point is similar to a border point but is not considered part of any cluster as the number of points within epsilon distance is lower than minimum points. Noise points are labelled as negative ones to distinguish them.

The clustering is performed using Google Colaboratory [53] using Python 3.10.12 [54] runtime on the Tensor Processing Unit v2 hardware accelerator. This was done to take advantage of this hardware accelerator's large amount of memory allocated, over 300 GBs. The Sklearn Python library version 1.2.2 implementation of DBSCAN [55] is used. It is chosen due to its ability to bulk-compute nearest neighbour calculations, significantly speeding up the algorithm at the cost of high memory usage. This is required to cluster millions of rows, and the memory usage is mitigated due to the large amount of memory provided by Google Collaboratory.

3.2.1 Minimum Points

DBSCAN requires two parameters to be set: minimum points and epsilon. Minimum points can be set to the dimensions of the data multiplied by two. The Latitude, Longitude, SOG, and COG are used since these values will be predicted later using the RNN and journey ID to group positions on the same journey together. Since the data has five dimensions, the minimum points should be set to ten. The minimum points value can then be used to find the optimum epsilon value.

3.2.2 Epsilon

Two experiments are performed to find the best Epsilon parameter: finding the elbow on the sorted k-distance graph and the results with the highest mean silhouette score.

The first experiment involves finding the Euclidean distance between the points and the minimum point number of nearest neighbours. Each distance is sorted from largest to smallest and plotted on a line graph. The elbow, the point where the decrease in the distance to the nearest neighbours slows down on the graph, indicates the range of best Epsilon values. The Epsilon values are from the distance on the y-axis.

The second experiment entails performing DBSCAN on a subset of the datasets using the range of Epsilon values and then calculating the mean silhouette score on the results. Silhouette score is calculated using equation (05) where a represents the average distance between a selected point and other points in the same cluster and b is the average distance between a selected point and all points in the nearest neighbouring cluster [20]. The silhouette score should produce a value between negative one and positive one. A higher silhouette score is better since it indicates the cluster is dense and well separated. The mean silhouette score is calculated by dividing all scores by the number of points. Due to the large number of rows being clustered, the silhouette score is calculated on a sample of points. The specific implementation of the silhouette score is from Sklearn [55] since it can be calculated using the mean and a sample of the clustering.

$$\text{silhouette score} = \frac{b-a}{\max(a,b)} \qquad (05)$$

3.3 AIS-based Prediction Model

As detailed in section 2.1.2, an RNN is used as the AIS-based prediction model. Before the RNN can be trained, the combined dataset must be split into the validation and training datasets. The training, validation, and testing datasets are all pre-processed to ensure they are formatted correctly for the RNN. The parameters of the RNN are decided through experiments to explore the best value.

The RNN training, prediction and experiments are done in Google Colaboratory [53] using the Python 3.10.12 [54] runtime with high RAM and V100 GPU hardware acceleration. The Keras library version 2.15.0 is used to create and train the RNN [29].

3.3.1 Splitting and Pre-processing

The combined positional data is split randomly with 40% of vessels for the validation dataset and 60% of vessels for the training dataset. They were randomly split to ensure no bias and to achieve better generalisation; the model is less likely to over-fit specific patterns. The ratio was picked to ensure the validation dataset has enough diversity to evaluate the model effectively. The testing dataset is not split; it is only used to test the model once it has finished training. The training dataset contains 2,037,108 rows, the validation dataset contains 997,591 rows, and the testing dataset contains 6,196,557 rows.

Before training, the data undergoes pre-processing to ensure it is correctly formatted. The sliding window method is used to format it. The x data contains a number of previous positions, and the y data contains the next position. This method allows the model to understand clearly how previous patterns in the data lead to the next position. The x data is then normalised using min-max normalisation, which ensures the data has a range between zero and one. This retains the original distribution of scores, except for the scaling factor. This reduces the predictive error from different scales [11].

3.3.2 Training and Prediction

As discussed in section 2.1.2, the RNN model comprises multiple BiGRU layers. BiGRU layers are used to improve model stability since vanishing and exploding gradients are prevented by BiGRUs. Since it has multiple hidden layers, it is deep. This should allow it to capture the complex patterns of ship movements. As shown in appendix K, the number of units begins at 1024 and decreases by a factor of two. This gradually reduces the dimensionality of the network, improving both the speed of training and regularisation by forcing the model to train on more generalisable data features in the lower layers. This regularisation will help to prevent over-fitting, which can be a problem with deep models. Over-fitting is where the model begins to train to the training data errors and noise rather than the general features [45]. If over-fitting is a problem, batch normalisation and early stopping can be employed to prevent this. Batch normalisation can help because it has a regularisation effect and reduces model instability, making the model generalise better [56]. Early stopping can help stop the model from training when over-fitting is detected. To ensure that the model has chance to train properly five hundred epochs are used to train the model.

In an exploratory analysis, different experiments are performed to find the optimum RNN parameters for each RNN column. The small-size dataset is used in these experiments since the smaller ships are more challenging to predict, hopefully distinguishing the best parameters. Further, it contains fewer ships, so the experiments are performed quickly.

The parameters being experimented on are the sliding window size, the optimisation function, the loss function, the batch size and the columns to include within the window. The experiments are performed within the same environment as training the actual RNN to ensure the results of the experiments can be replicated. To evaluate each experiment's success, the metrics are the predicted Mean Absolute Error (MAE) and Mean Squared Error (MSE). To act as a baseline, the parameters of window size of five, optimisation function of Adaptive Moment Estimation (Adam), loss function of MAE, batch size of 64 and no added columns. This baseline is used since these parameters should perform well with a low error. It is performed three times to ensure this baseline is accurate with the mean taken for each metric.

The different optimisation functions being experimented on are Adam, Adaptive Moment Estimation

Weighted (AdamW), Nesterov-accelerated Adaptive Moment Estimation (Nadam), AdaGrad (Adaptive Gradient) and Adadelta (Adaptive Delta) [29]. Variants of Adam are chosen due to their popularity and effectiveness [57] while AdaGrad and Adadelta are to ensure that other approaches are not more effective. The different loss functions being experimented on have to be suitable for regression. With that in mind, all the loss functions that Keras provides and are suitable are experimented on: MSE, MAE, Mean Squared Logarithmic Error (MSLE), Cosine Similarity, Huber and Log Cosh [29]. The batch size chosen will be every power of two, from the power of five to ten. The batch sizes are kept as powers of two as it often provides the best efficiency on hardware architectures [57]. The columns experimented with adding the journey identification number, clustering label, navigational status, heading, and turn rate, as they could provide extra information to aid in more accurate prediction. The sliding window size is to go from one to ten, incrementing by one for each experiment.

3.4 Testing

Testing is essential to verify that the AIS data is being pre-processed and analysed correctly. Each data pre-processing and analysis step has multiple unit tests to test each function and part of the steps. Each step also has an integration test to ensure that each function works correctly to produce the correct analysis or pre-processing. The integration tests involve using a test file with expected results and testing that the scripts achieve these results. The integration test involves running a test file through the entire data pre-processing and analysis step. Code coverage of these is used to evaluate how much of the code is being tested when tests are performed. Ideally, all code coverage is 100%, and as Table 1 shows, most of the code has been tested.

Processing Step	Code Coverage
Collection	78%
Combination	100%
Cleaning	100%
Repairing	98%
Journey Identification	100%
Journey Verification	100%
Clustering	100%
Splitting	100%
Training	100%
Prediction	100%

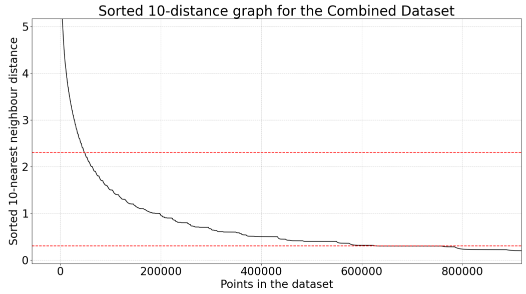
Table 1: Table showing the code coverage of different pre-processing and analysis steps in Python. The code coverage is generated using coverage library.

3.5 Exploratory Analysis

It is important to conduct exploratory data analysis to determine the appropriate parameters for both the DBSCAN clustering algorithm and the RNN model.

3.5.1 Clustering

As discussed in section 3.2.2, the elbow is identified on a k-distance graph to find the epsilon values range to experiment. The elbow is the transition point between the steep section and shallow section of the graph in between the red dotted lines. Since the minimum point is ten, it is a ten-distance graph. In Figure 3, the elbow can be shown as between 2.3 and 0.3 for the combined dataset. While in Figure 4, it can be shown between 2.6 and 0.05 for the test dataset. This provides a rough range of values to test and find the optimum epsilon value.



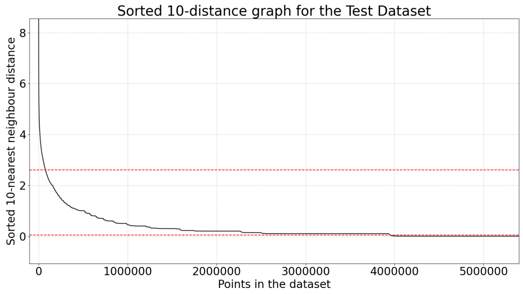
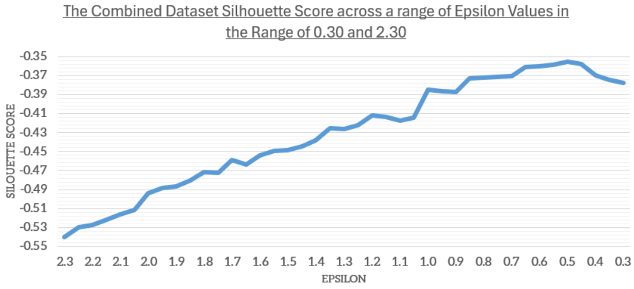


Figure 3 (left): This is a line graph of the sorted nearest ten distances to each point in the combined dataset. The top dotted red line is at y=2.3, and the bottom is at y=0.3, denoting the graph's elbow. This graph has been zoomed in to ensure the elbow is visible, so not all values are present.

Figure 4 (right): This is a line graph showing the sorted nearest ten distances to each point in the test dataset. The top dotted red line is at y=2.6, and the bottom is at y=0.05, denoting the elbow of the graph. This graph has been zoomed in to ensure the elbow is visible, so not all values are present.

Experimentation is conducted on the epsilon values in the range of the elbow of graphs in Figures 3 and 4. Each epsilon value is incremented by 0.05 between the ranges. The epsilon value with the highest silhouette score is 0.5, with a score of -0.3554 for the combined dataset and 0.1, with a score of -0.3276 for the test dataset. This is shown in Figures 5 and 6, respectively.



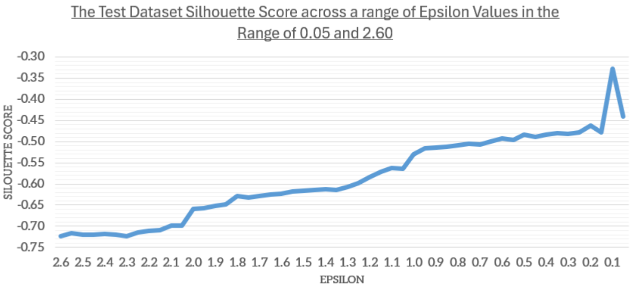


Figure 5 (left): This is a line graph of silhouette scores across a range of epsilon values for the combined dataset. For the full epsilon values and silhouette scores table, go to appendix I.

Figure 6 (right): This is a line graph of silhouette scores across a range of epsilon values for the test dataset. For the full epsilon values and silhouette scores table, go to appendix J.

3.5.2 RNN Experiments

As discussed in section 4.3.2, experiments are performed on the small dataset for each column. This is to find the most suitable parameters for the loss function, optimisation function, window size and any additional columns to include in each window.

3.5.3 Loss Function

The lowest prediction error through the MAE and MSE metrics determines the best loss function for each column. The MAE loss function was the most effective for the longitude and SOG columns. This was evident in the SOG column, where the predictive MAE is 0.3192 knots and an MSE of 3.8145 knots. This is lower than the next best loss function (Huber), with 0.4685 knots for MAE and 3.9703 knots for MSE. The choice of the best loss function was less apparent in the longitude column. However, the MAE loss function still outperformed the others, with a predictive MAE of 0.1455 degrees and MSE of 0.0287 degrees, slightly lower than the following best loss function of MSE with 0.1504 degrees for MAE and 0.0282 degrees for MSE. For the COG column, the best loss function is Log Cosh, as shown in Figure 7. Its predictions had an MAE of 0.6989 radians and an MSE of 1.5542 radians, with a similar MAE but a lower MSE than the next best, Huber. The loss function with the lowest predictive error for the latitude column is MSLE, as shown in Figure 8. Its predictions had an MAE of 0.0094 degrees and MSE of 0.0002 degrees, lower than the next best, Huber's MAE and MSE. For full tables, go to the appendices L, M, N and O.

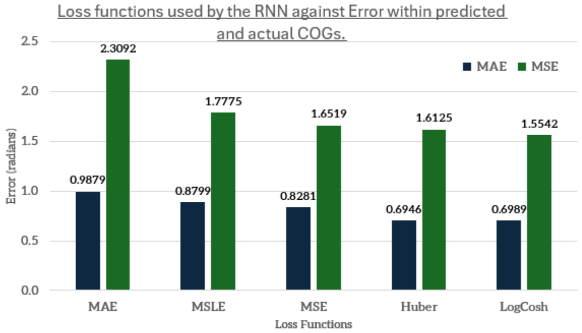


Figure 7 (left): This is a combined bar graph showing the MAE in blue and MSE in green of the different loss functions when predicting COGs.

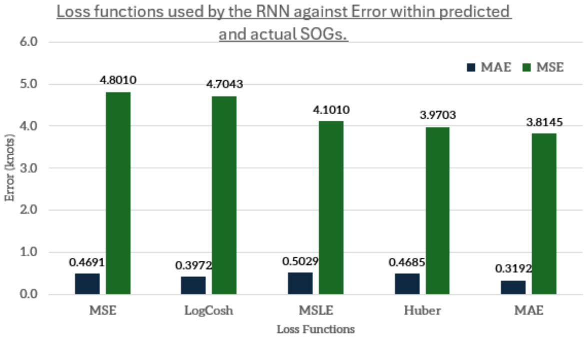


Figure 8 (right): This is a combined bar graph showing the MAE in blue and MSE in green of the different loss functions when predicting SOGs.

3.5.4 Optimisation Function

Similar to assessing the best loss functions, the lowest predictive error determines the best optimisation functions. For the COG column, the predictive errors did not clarify which function was best since the lower the MAE, the higher the MSE for all loss functions apart from Adam, as shown in Figure 9. The Nadam optimisation function was chosen since it had the lowest MAE and performed the best during training, with the lowest error on the final epoch and across average epochs. The AdamW optimisation function was the best for the latitude and SOG columns. It had the lowest predictive error for latitude, with an MAE of 0.0294 degrees and MSE of 0.0026 degrees, significantly lower than Adam, with 0.0378 degrees for MAE and 0.0045 degrees for MSE. It had the lowest predictive error for SOG, with an MAE of 0.2999 knots and MSE of 3.4251 knots, significantly lower than Adam, with 0.3192 knots for MAE and 3.8145 knots for MSE. The Adam optimisation function was best for the longitude column because it had the lowest error of 0.1455 degrees for MAE and 0.0287 degrees for MSE, as shown in Figure 10. This is lower than Nadam, with 0.1613 degrees for MAE and 0.0320 degrees for MSE.

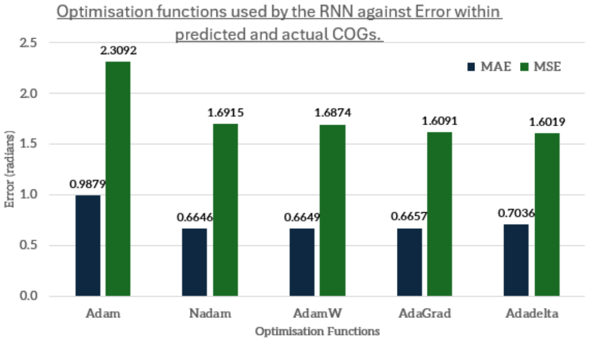


Figure 9 (left): This is a combined bar graph showing the MAE in blue and MSE in green of the different optimisation functions when predicting COGs.

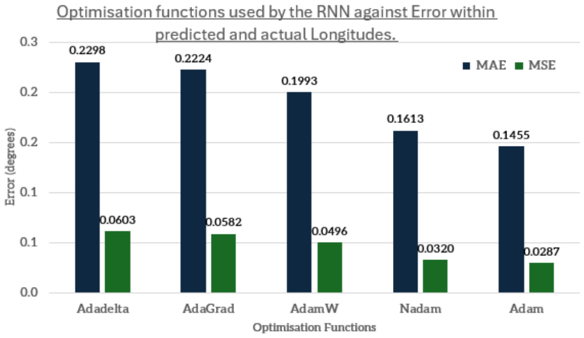


Figure 10 (right): This is a combined bar graph showing the MAE in blue and MSE in green of the different optimisation functions when predicting longitudinal coordinates.

3.5.5 Window Size

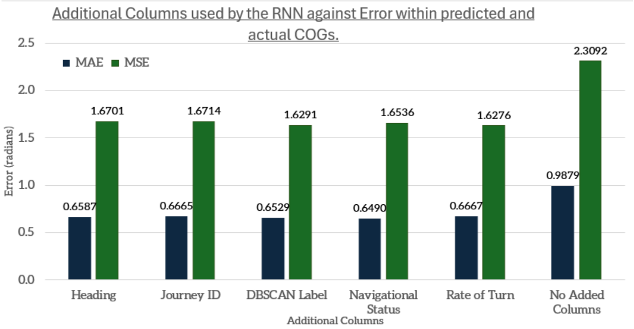
The window size affects the number of predicted positions, with journeys requiring at least a window size long to be predicted. Since this is the case, the window size must be the same for all columns. All columns' mean predictive MAE and MSE are taken to evaluate the best window size for all columns. A window size of seven is chosen since it has the lowest MSE of 1.2892 and the fourth lowest MAE of 0.3053. Window size of seven has a MSE 5% lower than the next best window size of six, even if it has a MAE 3.5% higher. Refer to appendix Q for the full table.

3.5.6 Batch Size

A variety of batch sizes were experimented on to find the most suitable. A batch size of 64 was chosen to ensure the error did not decrease significantly from the baseline while completing each epoch as quickly as possible. Across all columns, as batch size increased, the time taken to perform each epoch was reduced, with each increase resulting in a smaller time reduction. This was also the case for error; as the batch size was increased, predictive error generally increased. A batch size of 1024 was chosen since it reduced the time taken per epoch the most and did not significantly reduce error. Only column longitude had an excessively worse error for batch 1024 with a MSE 41.1% higher than batch 64, so batch size 512 was used as a MSE 18.8% higher than batch 64.

3.5.7 Additional Columns

Additional columns are added if they reduce the predictive MAE and MSE below the baseline with no additional columns. Navigational Status and DBSCAN label columns are added to the COG and SOG sliding window since they are shown to have significantly lower MAE and MSE than adding no columns as shown in Figure 11. The Rate of Turn column is added to the Latitude and Longitude sliding window since it produces a lower predictive MAE and MSE than adding no columns shown in Figure 12.



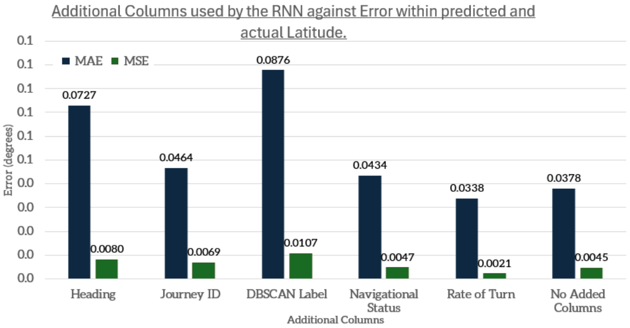


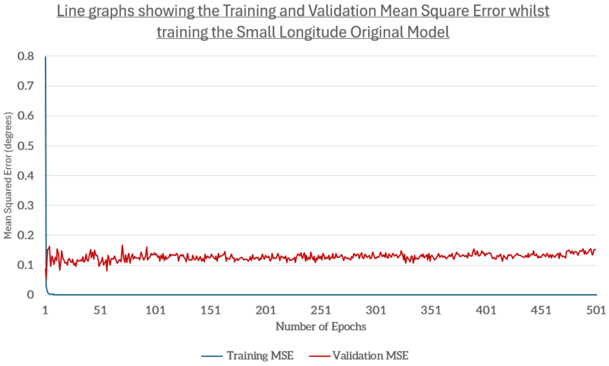
Figure 11 (left): This is a combined bar graph showing the MAE in blue and MSE in green of adding different columns to the sliding window when predicting COGs.

Figure 12 (right): This is a combined bar graph showing the MAE in blue and MSE in green of adding different columns to the sliding window when predicting the latitude coordinates.

3.5.8 RNN Predictions

The first time training using the original model, as shown in appendix K, all columns showed signs of over-fitting after five hundred epochs of training. Signs of over-fitting are a large disparity between the validation and training errors, the validation error not reducing whilst the training error is reducing, and sudden changes in validation error as the model. These sudden changes show that the model is no longer learning general patterns but the specific errors and patterns found within the training dataset.

When training for the small dataset's COG, SOG, and longitude columns, the models began over-fitting almost immediately. In contrast, the latitude column began over-fitting beyond the 295th epoch, as shown in Figure 14. Using longitude as an example, the training MSE decreased, reaching as low as 7.63E-05, but the validation MSE did not reflect the improvements the training MSE underwent. The validation MSE sharply increased after the first few epochs and then sporadically changed, as shown in Figure 13. Not only that but there is also a large difference between the validation MSE and training MSE, with a maximum difference of 0.1679 degrees. This is indicative of the model immediately over-fitting. Similar training occurred in the large dataset.



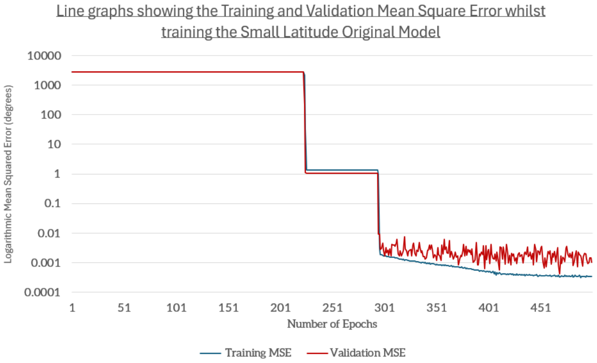


Figure 13 (left): A line graph showing the training error of the small longitude model using the original model. Validation MSE is shown in red while training MSE is shown in blue.

Figure 14 (right): This line graph shows the training error of the small latitude model using the original model. Since there is a large difference between the minimum and maximum MSE, a base-ten logarithmic scale is used for the MSE.

As discussed in section 3.3.2, there are several methods to mitigate over-fitting. Since over-fitting happens very early in the training, it indicates the model needs to be simplified for the data provided. To simplify the model, the first BiGRU is removed, and the number of units on the second BiGRU is halved. Batch normalisation is implemented between each of the BiGRUs. Not only that but L2 regularisation was also added to each of the BiGRU layers. This was done since regularisation introduces additional information in order to prevent overfitting. This new model is called the simplified model, while the old one is called the original model. The latitude column is trained and predicted using the original model, whilst the other columns use the simplified model. Early stopping is added for both models to prevent the model from training when the validation MAE does not decrease after twenty epochs. This would restore the model to the lowest weights. This is done on both models since even the original model began over-fitting for the latitude column.

4 Results

4.1 Invalid MMSI Numbers

No invalid MMSI number lengths were identified in either the positional or static testing dataset. However, there was one ship with an invalid MID in the positional and another thirteen in the static dataset. The one vessel (MMSI 102768367) with an invalid MID had an MID indicating that it was a search and rescue aircraft [58]. However, no search and rescue aircraft could be found within the ITU database with this MMSI number [59]. When validating the algorithm, it was able to achieve a 100% accuracy rate and a 0% false positive rate, as shown in Table 2.

Classification	# MMSI Numbers	# Correct	Correct (%)
Valid MID	50	50	100.00%
Invalid MID	50	50	100.00%

Table 2: A table showing the results from validating the algorithm to detect if the MID is valid.

4.2 Journey Verification

Once cleaned, the test static dataset contains 54,693 messages for 246 vessels. A vessel with no destination entered is an example of an invalid destination. With this in mind, there is a similar number of invalid and valid destinations, with 37.80% of static reports with invalid destinations and 37.36% of static reports with a valid destination, as shown in table 3. Nearly a quarter of the static messages could not be matched to a port. When validating the algorithm, this happened for various reasons: eight were outside the collection area, six were activities, not an actual destination, two were awaiting a destination, one had a spelling mistake, and three had unknown destinations. This shows that despite many failing to match a destination, it mostly happened for suitable reasons.

Classification	Number of rows	Percentage of rows
No Destination Entered	4,215	7.71%
Unable to match Destination	13,539	24.75%
Valid Destination	20,435	37.36%
Invalid Destination	16,504	30.18%

Table 3: A table showing the different classifications of the 54,693 static messages for the test datasets. No destination entered means that no destination string was entered. Unable to match the destination means the destination string could not be matched to a port. Valid destination means that the string destination matched the actual port it arrived in, and invalid means that these differ.

Using the equations from section 2.3, The accuracy of the destination detection is 83.48% with a false positive rate of 0.00% since no false positives occurred. In further investigation into the cause of the false negatives, seven of these were caused by missing positions. For example, the ship "Stena Europe" (MMSI 209649000) did go to the port in Fishguard, but the large number of missing positions resulted in the journey being incomplete. Another ten messages were classified as invalid because the actual journey destination was classified incorrectly, either by not finding any port or the wrong port. For example, the actual destination for the ship "Arco Dart" (MMSI 232003278) was classified as the "Mermaid Quay" rather than the port in "Cardiff" since it was moored closer to the "Mermaid Quay". "Mermaid Quay" was not in the ship's string destination, so it was identified as invalid. Two static messages were classified for an unknown reason, as it is not apparent what caused the incorrect classification. This indicates that

Classification	# Static Messages	# Vessels	# Correct	Correct (%)
No Destination Entered	5	4	5	100.00%
Unable to match Destination	20	19	20	100.00%
Valid Destination	70	35	51	72.86%
Invalid Destination	40	27	40	100.00%

Table 4: A table showing the results from validating that journey verification algorithm could correctly classify string destinations. The number of vessels shows the number of unique vessels in that category. The number correct is the number in that category that were classified as the same category by the destination verification.

4.3 Clustering

DBSCAN was used to cluster the data using the epsilon values in previous experiments. The combined dataset was clustered into 6,788 clusters with a silhouette score -0.3584. DBSCAN identified 365,411 noise points; each label contained an average of 446.94 positions for the combined dataset. The test dataset was clustered into 93,296 different clusters with a silhouette score of -0.3292. DBSCAN identified 3,367,942 noise points; each label contained an average of 66.41 positions for the test dataset.

4.4 RNN Prediction

Implementing the new model for the COG, SOG, and Longitude columns has significantly reduced the validation error. The large COG MSE, initially at 223.7331 degrees, has been dramatically reduced to 2.8661 degrees. Similarly, the small Longitude, which started at 0.1516 degrees, has been brought down to 0.0274 degrees. This reduction in error shows that the improvements made in the simplified model have worked. Prior literature suggested that the large dataset should have been easier to predict and, therefore, have a lower error. However, as Table 5 shows, the large dataset appeared to have a higher error than the small dataset.

Columns and Size	MAE	MSE
Small Latitude	0.0236 degrees	0.0044 degrees
Small Longitude	0.1752 degrees	0.0414 degrees
Small COG	13.3126 degrees	14.8666 degrees
Small SOG	0.2902 knots	0.5323 knots
Large Latitude	1.6848 degrees	3.6832 degrees
Large Longitude	0.0601 degrees	0.0052 degrees
Large COG	5.3150 degrees	9.4962 degrees
Large SOG	6.4998 knots	72.4881 knots

Table 5: A table showing the predictive MAE and MSE for each dataset. The predictions are made using the small and large test dataset. The simplified models are used for predicting COG, SOG and longitude while the original model is

used when predicting the latitude.

As shown in Table 6, every model detected many anomalies. This number of anomalies is far higher than expected, except for the large COG anomalies. This indicates that the models did not perform well and detected too many anomalies. This appears to be especially the case for the distance anomalies for the large dataset. This suggests the model's predictions are very incorrect and failed to learn the general patterns within the training dataset.

Dataset	Anomaly	Number of Anomalies	Percentage of Anomalies
Small	Large Distance Anomaly	3,467,695	99.85%
Small	Small Distance Anomaly	4,409	0.13%
Small	COG Anomaly	2,188,519	63.02%
Small	SOG Anomaly	3,313,741	95.42%
Large	Large Distance Anomaly	2,721,750	100.00%
Large	Small Distance Anomaly	0	0.00%
Large	COG Anomaly	765,520	28.13%
Large	SOG Anomaly	2,709,360	99.54%

Table 6: This table shows the number of anomalies detected within the test small and large datasets. Each position can be assigned either a small or large distance anomaly, a COG anomaly, or a SOG anomaly. The number of positions within the small dataset is 3,472,756, and it is 2,721,752 for the large dataset.

To assess the models' performance, the anomalies the model identified in the test dataset are compared to anomalies identified by a human. The anomalies are identified by looking for sudden changes in the data or by looking at all values at that time to see if it is impossible (e.g., ships cannot turn as fast when travelling fast).

Two five-hundred-point journeys were used to determine the performance of each of the small and large models. One journey was chosen because it was more regular, and the other because the ship behaved anomalously. For example, the large model was evaluated against a journey of the 'Stena Europe' (MMSI 209649000) on Friday, March 8th, 2024, between 6:38:21 PM and 7:12:13 PM as shown in Figure 15. This specific journey was chosen because the ship begins turning sharply. It was also evaluated against a journey of the 'Beatriz B' (MMSI 209056000) on Sunday, March 10th, 2024, between 8:11:24 AM and 9:32:22 AM. This specific journey was chosen since the ship travelled more normally. The results of these evaluations are shown in Table 7.

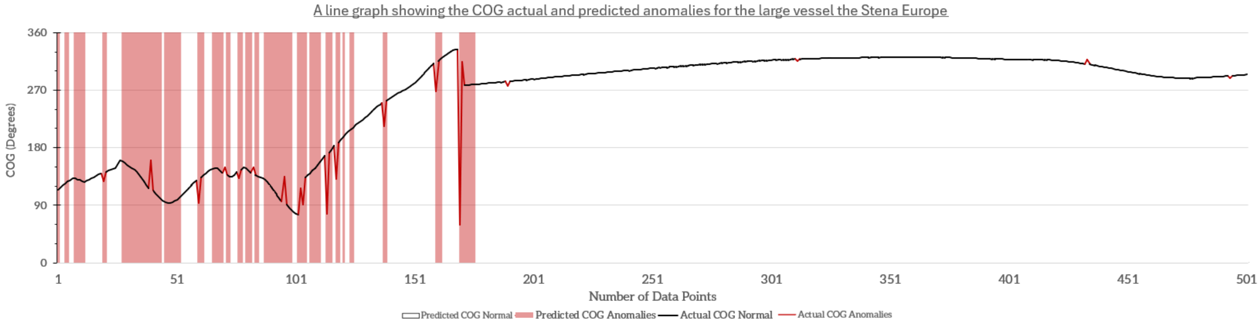


Figure 15: This is a line graph showing the actual COG, with actual anomalies in the COG highlighted by a red line. The predicted COG anomalies are shown by the red background.

Dataset	Column	Average Accuracy	Average False Positive
Large	Distance	1.8%	97.9%
Large	COG	85.4%	13.0%
Large	SOG	2.2%	97.8%
Small	Distance	10.2%	89.8%
Small	COG	60.3%	39.1%
Small	SOG	14.2%	85.6%

Table 7: A table showing the results from validating that journey verification algorithm could correctly classify string destinations. The number of vessels shows the number of unique vessels in that category. The number correct is the number in that category that were classified as the same category by the destination verification.

To evaluate the model further, the model is also predicted on a dataset gathered between 2024-04-06 and 2024-04-09. This dataset was gathered when several storms were passing between the Irish Sea.

Comparing Table 8 to Table 6, there is an increase in all anomalies apart from large COG anomalies. For example, the small COG anomalies increased by 6.2% while large COG anomalies decreased by 9.5%. This indicates that the models roughly detect anomalies, although a far larger change would be expected if it were detecting them well.

Dataset	Anomaly	Number of Anomalies	Percentage of Anomalies
Small	Large Distance Anomaly	1,621,065	100.00%
Small	Small Distance Anomaly	0	0.00%
Small	COG Anomaly	1,089,532	67.21%
Small	SOG Anomaly	1,619,710	99.92%
Large	Large Distance Anomaly	1,999,769	100.00%
Large	Small Distance Anomaly	0	0.00%
Large	COG Anomaly	514,621	25.73%
Large	SOG Anomaly	1,996,066	99.81%

Table 8: This table shows the number of anomalies detected when storms were passing through the collection area. Each position can be assigned either a small or large distance anomaly, a COG anomaly, or a SOG anomaly. The number of positions within the small dataset is 1,621,065, and it is 1,999,769 for the large dataset.

5 Discussion

This paper focused on the detection of positional, kinematic and data-related anomalies in the shipping activity between Ireland and Wales. This aimed to improve MDA in this area. An RNN was created to predict ship movements to detect positional and kinematic anomalies to achieve this. Furthermore, missing and delayed vessel positions were detected using the ship's nominal reporting intervals, which were then repaired with Cubic Hermite Spline Interpolation, and incorrect journey destinations were detected. These processes were carried out to detect data-related anomalies and improve the detection of positional and kinematic anomalies. All anomalies were detected using the testing positional and static reports captured in early March 2024 with additional port information from the WPI and OSM. Hypotheses one, two and three failed. Hypotheses one and two were that positional and kinematic anomalies could be detected with an accuracy of 96.35% or greater and a false alarm rate of 1.67% or lower. Hypotheses three and four are that incorrect and valid journeys could be detected with an accuracy of 84.1% or greater and a false positive rate of 2%. Hypothesis five is that using DBSCAN improves the detection of positional and kinematic anomalies. Hypotheses one, two and three failed while hypothesis five only partially succeeded. Hypotheses four, six and seven succeeded.

Hypotheses one and two failed because the positional reports were too degraded. Only 6.29% and 4.74% of positions were reported at the nominal reporting interval in the combined and test datasets. This was far lower than the expected percentage of 42.3% reported in [38]. Therefore, significant repair to the positional data utilising Cubic Hermite Spline Interpolation was needed. However, as [40] highlighted, Cubic Hermite Spline Interpolation is less effective when the data is excessively degraded. In addition, [11] highlighted that to achieve the 96.35% accuracy and 1.67% false alarm rate, high-quality positional data is required, which was not the case with the positional data from [49].

Hypothesis three could have failed because the port dataset is inaccurate. The WPI had incorrect positions for the ports. It is difficult to determine the exact centre of a port, but this is not the case with Bangor, Northern Ireland, which is 1.63 kilometres from the actual centre of the port. Many ports are missing, with oil terminals around Pembroke, Wicklow, Ballycastle, and Church Bay ports not included on the WPI or OSMs. This causes the journey verification to fail since it cannot find the port to verify the journey. It should be noted that hypothesis three only failed by a small margin with less than a one per cent difference in accuracy, so an accurate port dataset could have resulted in a higher accuracy than the hypothesis.

Hypothesis five partially succeeded since clustering the positional data with DBSCAN improved the detection of kinematic anomalies, not positional anomalies. This is because adding the DBSCAN labels to the sliding window for columns COG and SOG was shown to reduce the error but not for latitude or longitude columns. For example, using DBSCAN labels reduced the predictive MSE by

11.53% for the SOG column and by 28.39% for the COG column. In contrast, using DBSCAN labels increased the predictive MSE by 137.78% for the latitude column and by 5.58% for the longitude column. Further, the DBSCAN clustering could have been improved by further exploratory analysis into the sigma and minimum points values, with both datasets having a silhouette score that indicated poor fit.

6 Limitations

The main limitation of this paper is the missing AIS and port data. The missing data likely resulted in failing hypotheses one, two, and three. The AIS positional data was incomplete because it was out of range of the AIS stations. This was made worse by the insufficient number of ground stations in the collection area, with data only from 37 stations used [49] opposed to the 113 in the same area used for [60]. Since only data from ground stations is included, any messages outside their range will be missing. Satellite AIS positional data could have verified the ground AIS positional data and provided access to messages outside the ground station range. As discussed previously, the port dataset was missing a number of ports; the journey verification algorithm would have benefitted from having a more complete dataset.

Another limitation is transmission errors in the AIS data. These cause motion logic errors that result in sudden changes in speed and position, as shown in [11]. Such sudden changes result in the RNN model being unable to generalise as well since it trains to these errors that are not reflective of vessel movements. For example, COG could be predicted more accurately than SOG and positions. This could have been overcome using the average speed and position to correct the errors [11].

Another limitation is the inclusion of redundant positional data. Redundant positional data likely made it easier for the RNN to overfit the training data, resulting in the early stopping triggering earlier and the model having a higher validation loss. Removing redundant data could have allowed for a larger collection area or a longer collection period for the AIS training data, which could have made the model able to generalise better. The Douglas–Poker algorithm could have been used to compress the positional data to remove redundant data [61].

7 Conclusion

In conclusion, despite their limitations, the models and algorithms in this paper hold significant potential for detecting anomalous vessel behaviour. The implication of this has the potential to improve the UK's MDA and ensure the Irish Sea's security. One of the key findings of this paper is that with more complete datasets and more robust cleaning methods, these models could reach a point where their implementation provides practical results for the police or shipping industry. For example, the models and algorithms could be combined to create a real-time anomaly detection website.

Future research would benefit from having vessel anomalies classified in a dataset. This dataset would be classified by those knowledgeable about what would constitute a vessel anomaly. A dataset classified this way would provide ground truths to the model, making the anomalies identified by the model comparable to those detected by the expert. Furthermore, making this dataset available for other research would allow for better comparison between different models and replication of results.

In addition, although this paper has covered a comprehensive set of differing anomalous vessel behaviours, detecting further categories of anomalies would allow a complete picture of anomalous activity. Deliberately turning off AIS transponders allows one to hide activity from AIS. Therefore, the detection of this would be beneficial for the identification of security threats. Further, all static and voyage-related AIS data could be verified against an external database. This process would be beneficial for the detection of discrepancies with this data.

References

International Maritime Organization, "Enhancing maritime domain awareness in West Indian Ocean and Gulf of Aden." https://www.imo.org/en/MediaCentre/Pages/WhatsNew-1203.aspx, lastaccessed=2023-11-14, 2018.

European Commission, "Towards the integration of maritime surveillance: A common information sharing environment for the eu maritime domain." https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2009:0538:FIN:EN:PDF, lastaccessed=2024-03-04, 2009.

Department for Transport, "Port freight annual statistics 2022: Arrivals," https://www.gov.uk/government/statistics/port-freight-annual-statistics-2022/port-freight-annual-statistics-2022-arrivals, lastaccessed=2024-03-17, 2023.

Parliamentary Office of Science & Technology, "Marine planning," https://www.parliament.uk/globalassets/documents/post/postpn388_Marine-Planning.pdf, lastaccessed=2023-11-14, 2011.

E. Stebbings, E. Papathanasopoulou, T. Hooper, M. C. Austen, and X. Yan, "The marine economy of the United Kingdom," *Marine Policy*, vol. 116, p. 103905, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308597X19307390

Department for Transport, "National strategy for maritime security," https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1100525/national-strategy-for-maritime-security-web-version.pdf, lastaccessed=2023-11-14, 2022.

G. Duparc-Portier and G. Figus, "The impact of the new Northern Ireland protocol: can Northern Ireland enjoy the best of both worlds?" *Regional Studies*, vol. 56, no. 8, pp. 1404–1417, 2022. [Online]. Available: https://doi.org/10.1080/00343404.2021.1994547

Marine Management Organisation, "Mapping UK shipping density and routes from AIS," https://assets.publishing.service.gov.uk/media/5a7edadce5274a2e87db254b/1066.pdf, lastaccessed=2024-03-03, 2014.

M. Riveiro, G. Pallotta, and M. Vespe, "Maritime anomaly detection: A review," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 5, p. e1266, 2018. [Online]. Available: https://wires-onlinelibrary-wiley-com.uoelibrary.idm.oclc.org/doi/abs/10.1002/widm.1266

D. Zhang, J. Li, Q. Wu, X. Liu, X. Chu, and W. He, "Enhance the AIS data availability by screening and interpolation," in *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, 2017, pp. 981–986.

B. Zhang, K. Hirayama, H. Ren, D. Wang, and H. Li, "Ship anomalous behavior detection using clustering and deep recurrent neural network," *Journal of Marine Science and Engineering*, vol. 11, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2077-1312/11/4/763

T. Emmens, C. Amrit, A. Abdi, and M. Ghosh, "The promises and perils of automatic identification system data," *Expert Systems with Applications*, vol. 178, p. 114975, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421004164

International Telecommunication Union Radiocommunication Sector, "Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band," https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1371-5-201402-I!!PDF-E.pdf, lastaccessed=2024-03-29, 2014.

International Maritime Organization, "Revised guidelines for the onboard operational use of shipborne automatic identification systems (AIS)," https://wwwcdn.imo.org/localresources/en/OurWork/Safety/Documents/AIS/Resolution%20A.1106(29).pdf, lastaccessed=2024-03-17, 2015.

International Telecommunication Union Radiocommunication Sector, "Assignment and use of identities in the maritime mobile service," https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.585-9-202205-I!!PDF-E.pdf, lastaccessed=2024-03-29, 2022.

A. Sidibé and G. Shu, "Study of automatic anomalous behaviour detection techniques for maritime vessels," *Journal of Navigation*, vol. 70, no. 4, p. 847–858, 2017.

L. Zhao and G. Shi, "Maritime anomaly detection using density-based clustering and recurrent neural network," *The Journal of Navigation*, vol. 72, no. 4, p. 894–916, 2019.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

L. Rokach and O. Maimon, "Clustering methods," *Data mining and knowledge discovery handbook*, pp. 321–352, 2005.

P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

S. Mehri, A. A. Alesheikh, and A. Basiri, "A contextual hybrid model for vessel movement prediction," *IEEE Access*, vol. 9, pp. 45 600–45 613, 2021.

H. Zhu, M. Akrout, B. Zheng, A. Pelegris, A. Jayarajan, A. Phanishayee, B. Schroeder, and G. Pekhimenko, "Benchmarking and analyzing deep neural network training," in *2018 IEEE International Symposium on Workload Characterization (IISWC)*, 2018, pp. 88–100.

J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities." *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, 1998. [Online]. Available: https://doi.org/10.1142/S0218488598000094

P. Khandelwal, J. Konar, and B. Brahma, "Training RNN and it's variants using sliding window technique," in *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, 2020, pp. 1–5.

I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

F. Chollet *et al.*, "Keras 2.15.0," https://keras.io, lastaccessed=2024-04-10, 2015.

N. Zakaria, "Effect of ship size, forward speed and wave direction on relative wave height of container ships in rough seas," *Journal of the Institution of Engineers*, vol. 72, no. 3, pp. 21–34, 2009.

S. Inoue, M. Hirano, K. Kijima, and J. Takashina, "A practical calculation method of ship maneuvering motion," *International Shipbuilding Progress*, vol. 28, no. 325, pp. 207–222, 1981.

G. Rutkowski and A. Królikowski, "Simplified method for estimating maximum ship's draught when navigating in shallow water on the south of stolpe bank in the aspect of the vessels with maximum dimensions and draught," *TransNav, International Journal on Marine Navigation and Safety od Sea Transportation*, vol. 4, no. 4, 2010.

C. C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.

C. Iphar, A. Napoli, C. Ray, E. Alincourt, and D. Brosset, "Risk Analysis of falsified Automatic Identification System for the improvement of maritime traffic safety," in *ESREL 2016*, L. Walls, M. Revie, and T. Bedford, Eds. Glasgow, United Kingdom: Taylor & Francis, Sep. 2016, pp. 606–613 – ISBN 978–1–138–02 997–2. [Online]. Available: https://minesparis-psl.hal.science/hal-01421905

I. Kontopoulos, K. Chatzikokolakis, D. Zissis, K. Tserpes, and G. Spiliopoulos, “Real-time maritime anomaly detection: detecting intentional AIS switch-off,” *International Journal of Big Data Intelligence*, vol. 7, no. 2, pp. 85–96, 2020. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJBDI.2020.107375

F. Mazzarella, M. Vespe, A. Alessandrini, D. Tarchi, G. Aulicino, and A. Vollero, “A novel anomaly detection approach to identify intentional AIS on-off switching,” *Expert Systems with Applications*, vol. 78, pp. 110–123, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417417300933

M. Liang, R. W. Liu, Q. Zhong, J. Liu, and J. Zhang, “Neural network-based automatic reconstruction of missing vessel trajectory data,” in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, 2019, pp. 426–430.

P. Last, C. Bahlke, M. Hering-Bertram, and L. Linsen, “Comprehensive analysis of automatic identification system (AIS) data in regard to vessel movement prediction,” *Journal of Navigation*, vol. 67, no. 5, p. 791–809, 2014.

C. L. Bajaj, “Multi-dimensional hermite interpolation and approximation for modelling and visualization.” in *ICCG*, 1993, pp. 335–348.

B. Zaman, D. Marijan, and T. Kholodna, “Interpolation-based inference of vessel trajectory waypoints from sparse AIS data in maritime,” *Journal of Marine Science and Engineering*, vol. 11, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2077-1312/11/3/615

D. Yang, L. Wu, and S. Wang, “Can we trust the AIS destination port information for bulk ships?–implications for shipping policy and practice,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 149, p. 102308, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136655452100082X

K. Wolsing, L. Roepert, J. Bauer, and K. Wehrle, “Anomaly detection in maritime AIS tracks: A review of recent approaches,” *Journal of Marine Science and Engineering*, vol. 10, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2077-1312/10/1/112

A. McAbee, J. Scrofani, M. Tummala, D. Garren, and J. McEachen, “Traffic pattern detection using the hough transformation for anomaly detection to improve maritime domain awareness,” in *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1–6.

J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, “On mean absolute error for deep neural network based vector-to-vector regression,” *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020.

G. James, D. Witten, T. Hastie, and R. Tibshirani, *Statistical Learning.* New York, NY: Springer US, 2021, pp. 15–57. [Online]. Available: https://doi.org/10.1007/978-1-0716-1418-1_2

G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *Multiple Testing.* Cham: Springer International Publishing, 2023, pp. 557–596. [Online]. Available: https://doi.org/10.1007/978-3-031-38747-0_13

G. van Rossum, “Python Programming Language 3.11.5,” https://www.python.org/, lastaccessed=2024-04-10, 1991.

T. pandas development team, “pandas-dev/pandas: Pandas 2.1.1,” 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

AIS-Stream, “AIS-Stream.io,” https://aisstream.io, lastaccessed=2023-11-18, 2022.

National Geospatial-Intelligence Agency, “World port index,” https://msi.nga.mil/Publications/WPI, lastaccessed=2023-11-18, 2019.

S. Coast, “Open street maps,” www.openstreetmap.org, lastaccessed=2023-11-18, 2004.

United Nations Economic Commission for Europe, “Un/locode,” https://unece.org/trade/uncefact/unlocode, lastaccessed=2024-04-13, 2023.

Google LLC, “Google Colaboratory,” https://colab.research.google.com/, lastaccessed=2024-04-10, 2017.

G. van Rossum, “Python Programming Language 3.10.12,” https://www.python.org/, lastaccessed=2024-04-10, 1991.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, “Understanding batch normalization,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/36072923bfc3cf47745d704feb489480-Paper.pdf

C. C. Aggarwal, *Training Deep Neural Networks.* Cham: Springer International Publishing, 2018, pp. 105–167. [Online]. Available: https://doi.org/10.1007/978-3-319-94463-0_3

International Telecommunication Union Radiocommunication Sector, “Table of maritime identification digits,” https://www.itu.int/en/ITU-R/terrestrial/fmd/Pages/mid.aspx, lastaccessed=2024-04-01, 2022.

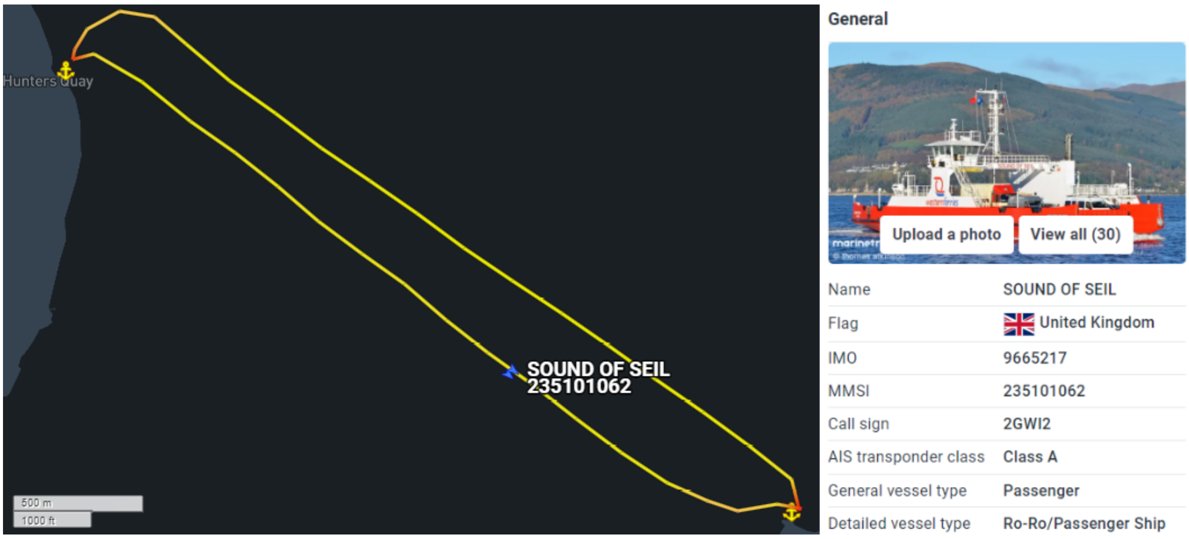
International Telecommunication Union, “Search and rescue aircraft database,” https://www.itu.int/mmsapp/SearchAndRescueAircraft/list, lastaccessed=2024-04-30, 2022.

D. Lekkas, “Marine traffic,” https://www.marinetraffic.com/en/ais/home/, lastaccessed=2023-11-17, 2006.

D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

Appendices

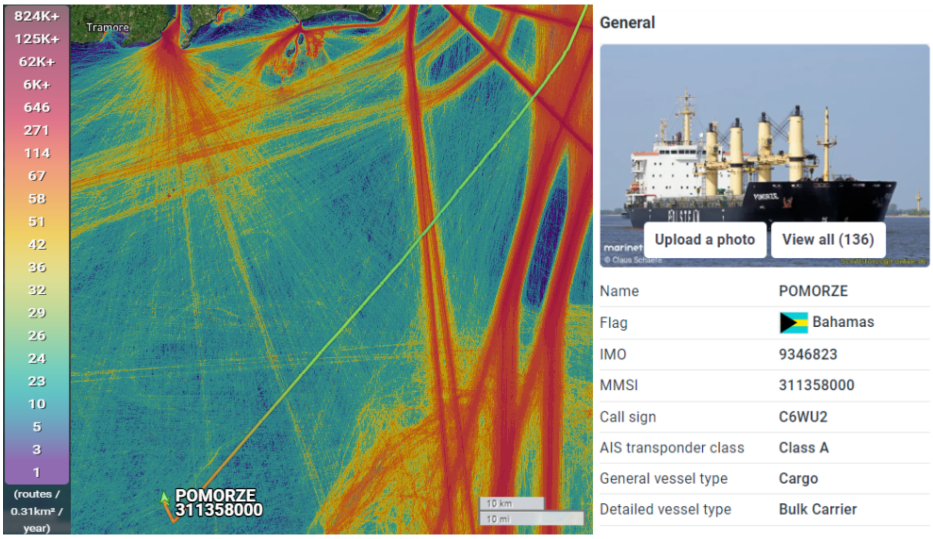
A Normal Vessel Behaviour Example



Name	SOUND OF SEIL
Flag	United Kingdom
IMO	9665217
MMSI	235101062
Call sign	2GWI2
AIS transponder class	Class A
General vessel type	Passenger
Detailed vessel type	Ro-Ro/Passenger Ship

Appendix A: A map of the mouth of the River Clyde with the line showing a journey of the Ro-Ro vessel ”Sound of Seil” between Mcinroy port and Hunters quay. Additional vessel information is shown on the right. This journey is normal since it is a typical journey that the ship performs and does not abruptly change direction or speed. The line's colour shows the vessel's speed, going from red (stopped) to green (max speed). It was captured on 17/11/2023 from the marine traffic website [60].

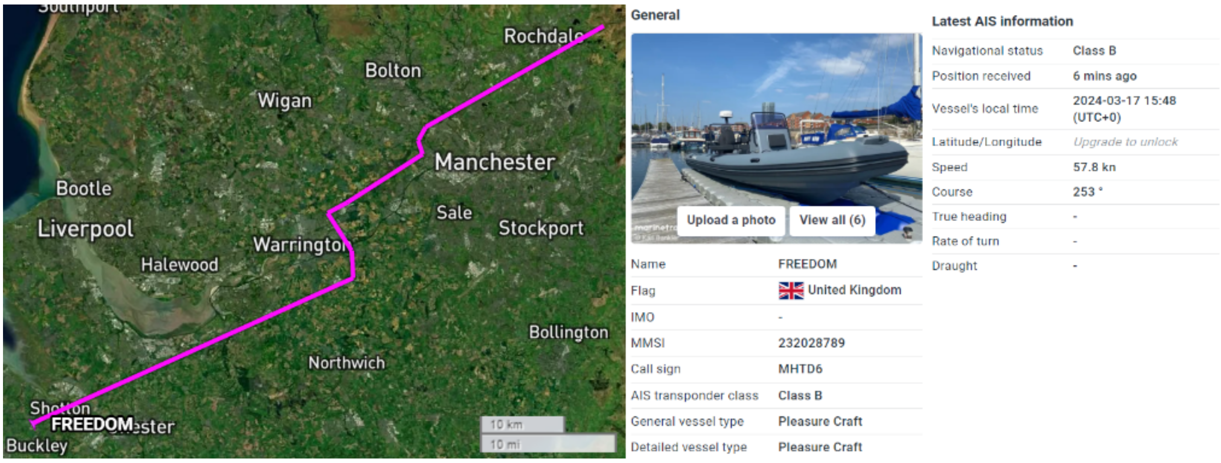
B Positional and Kinematic Anomaly Example



Name	POMORZE
Flag	Bahamas
IMO	9346823
MMSI	311358000
Call sign	C6WU2
AIS transponder class	Class A
General vessel type	Cargo
Detailed vessel type	Bulk Carrier

Appendix B: A density map of vessels travelling in 2022 with the line showing the journey of the cargo vessel ”Pomorze”. Additional vessel information is shown on the right. The density map goes from blue (lowest density) to red (highest density), as shown by the scale on the left. The line's colour shows the vessel's speed, going from red (stopped) to green (max speed). The vessel's journey is unusual since it moves away from the busy shipping lanes (the dark red areas on the map), indicating a positional anomaly. Then, it slows down and takes a sharp turn north, indicative of a kinematic anomaly. It was captured on 17/11/2023 from the marine traffic website [60].

C Data-related Anomaly Example



General

Name	FREEDOM
Flag	United Kingdom
IMO	-
MMSI	232028789
Call sign	MHTD6
AIS transponder class	Class B
General vessel type	Pleasure Craft
Detailed vessel type	Pleasure Craft

Latest AIS information

Navigational status	Class B
Position received	6 mins ago
Vessel's local time	2024-03-17 15:48 (UTC+0)
Latitude/Longitude	*Upgrade to unlock*
Speed	57.8 kn
Course	253 °
True heading	-
Rate of turn	-
Draught	-

Appendix C: A satellite map with the line showing the journey of the pleasure craft ”Freedom”. Additional vessel information is shown on the right. The journey is irrational since the ship travelled at 57.8 knots (66.5 miles per hour) on land and followed roads. The AIS transmitter was likely left on while being towed, resulting in the data-related anomaly. It was captured on 17/03/2024 from the marine traffic website [60].

D Table of the Nominal Reporting Intervals

Ship’s dynamic conditions	**Nominal reporting interval**
Ship at anchor or moored and SOG $\leq$ 3 knots	3 minutes
Ship at anchor or moored and SOG $>$ 3 knots	10 seconds
SOG 0-14 knots	10 seconds
SOG 0-14 knots and changing course	3 1/3 seconds
SOG 14-23 knots	6 seconds
SOG 14-23 knots and changing course	2 seconds
SOG $\geq$ 23 knots	2 seconds
SOG $\geq$ 23 knots and changing course	2 seconds

Appendix D: The nominal reporting interval for Class A ship-borne mobile equipment [13]. The ship’s size, speed, and rate of turning determine whether a ship is considered to be changing course. Ship at anchor or moored refers to the corresponding navigation status.

E Table of Missing Position Values for the Combined Dataset

Column Name	**Number Missing**	**Percentage Missing**
Navigational Status	29,249	2.07%
Rate of Turning	166,116	11.75%
Sog	297	0.02%
Latitude	0	0.0%
Longitude	0	0.0%
Cog	100,982	7.14%
Heading	159,649	11.29%

Appendix E: A table showing the number of missing position values and out of range values for each column in the combined dataset. The percentages are calculated using the original number of rows (before dropping any rows), which is 1,413,520 rows.

F Table of Missing Position Values for the Test Dataset

Column Name	Number Missing	Percentage Missing
Navigational Status	24,707	2.22%
Rate of Turning	138,680	12.48%
Sog	522	0.05%
Latitude	0	0.0%
Longitude	0	0.0%
Cog	94,369	8.49%
Heading	142,516	12.83%

Appendix F: A table showing the number of missing position values and out of range values for each column in the test dataset. The percentages are calculated using the original number of rows (before dropping any rows), which is 1,111,221 rows.

G Table of Categories of Delayed Combined Dataset

Delay Categories	Number in Category	Percentage in Category
Normal	10,583	6.29%
Delayed	12,096	7.19%
Needs Repair	145,309	86.42%
Cannot be Repaired	147	0.09%

Appendix G: A table showing the number of positions from the combined dataset categorised by how delayed they are from the previous position. This only shows the original dataset before it has been repaired but after removing duplicates, large time gaps and missing values, which contains 168,135 rows.

H Table of Categories of Delayed Test Dataset

Delay Categories	Number in Category	Percentage in Category
Normal	19,007	4.74%
Delayed	21,877	5.46%
Needs Repair	359,904	89.77%
Cannot be Repaired	118	0.03%

Appendix H: A table showing the number of positions from the test dataset categorised by how delayed they are from the previous position. This only shows the original dataset before it has been repaired but after removing duplicates, large time gaps and missing values, which contains 400,906 rows.

I Table of silhouette scores after clustering across the range of epsilon values in the combined dataset

Epsilon	silhouette Score
2.30	-0.5404
2.25	-0.5301
2.20	-0.5274
2.15	-0.5217
2.10	-0.5164
2.05	-0.5117
2.00	-0.4936
1.95	-0.4882
1.90	-0.4866
1.85	-0.4804
1.80	-0.4718
1.75	-0.4726
1.70	-0.4591
1.65	-0.4638
1.60	-0.4541
1.55	-0.4492
1.50	-0.4484
1.45	-0.4447
1.40	-0.4385
1.35	-0.4257
1.30	-0.4260
1.25	-0.4219
1.20	-0.4120
1.15	-0.4132
1.10	-0.4173
1.05	-0.4146
1.00	-0.3845
0.95	-0.3862
0.90	-0.3873
0.85	-0.3727
0.80	-0.3721
0.75	-0.3711
0.70	-0.3709
0.65	-0.3614
0.60	-0.3604
0.55	-0.3587
0.50	**-0.3554**
0.45	-0.3575
0.40	-0.3694
0.35	-0.3744
0.30	-0.3780

Appendix I: This is a table showing the resulting silhouette score after performing DBSCAN on the combined dataset for the range of epsilon values. The best silhouette score is highlighted in bold.

J Table of silhouette scores after clustering across the range of epsilon values in the test dataset

Epsilon	**silhouette Score**
2.60	-0.7241
2.55	-0.7162
2.50	-0.7200
2.45	-0.7200
2.40	-0.7189
2.35	-0.7193
2.30	-0.7230
2.25	-0.7140
2.20	-0.7109
2.15	-0.7097
2.10	-0.6991
2.05	-0.6992
2.00	-0.6584
1.95	-0.6576
1.90	-0.6526
1.85	-0.6488
1.80	-0.6293
1.75	-0.6326
1.70	-0.6293
1.65	-0.6256
1.60	-0.6225
1.55	-0.6170
1.50	-0.6166
1.45	-0.6143
1.40	-0.6117
1.35	-0.6149
1.30	-0.6064
1.25	-0.5983
1.20	-0.5841
1.15	-0.5716
1.10	-0.5629
1.05	-0.5650
1.00	-0.5297
0.95	-0.5151
0.90	-0.5139
0.85	-0.5125
0.80	-0.5091
0.75	-0.5048
0.70	-0.5064
0.65	-0.5001
0.60	-0.4925
0.55	-0.4963
0.50	-0.4844
0.45	-0.4898
0.40	-0.4842
0.35	-0.4802
0.30	-0.4809
0.25	-0.4781
0.20	-0.4629
0.15	-0.4776
0.10	**-0.3276**
0.05	-0.4408

Appendix J: This is a table showing the resulting silhouette score after performing DBSCAN on the test dataset for the range of epsilon values. The best silhouette score is highlighted in bold.

K A table showing the structure of the original RNN

Layer Type	Output Shape	Number of Parameters
Bidirectional GRU	(None, 7, 1024)	1,597,440
Bidirectional GRU	(None, 7, 512)	1,969,152
Bidirectional GRU	(None, 7, 256)	493,056
Bidirectional GRU	(None, 7, 128)	123,648
Bidirectional GRU	(None, 64)	31,104
Dense	(None, 16)	1,040
Dense	(None, 8)	136
Dense	(None, 1)	9

Appendix K: Table showing the design and structure of the original RNN. It shows the different layers shape and type within the RNN. Generated using the summary function in Keras [29].

L A table showing the predictive loss when experimenting on different loss function for the COG column

Loss Function	MAE (radians)	MSE (radians)
Cosine Similarity	53.5948	2875.2276
MAE	0.9879	2.3092
MSLE	0.8799	1.7775
MSE	0.8281	1.6519
Huber	0.6946	1.6125
Log Cosh	0.6989	1.5542

Appendix L: A table showing the predictive loss across different loss functions for the COG column.

M A table showing the predictive loss when experimenting on different loss function for the latitude column

Loss Function	MAE (degrees)	MSE (degrees)
Cosine Similarity	38.6807	1497.3213
MAE	19.3946	377.1541
MSLE	0.0184	0.0010
MSE	0.0094	0.0002
Huber	51.8493	2688.3487
Log Cosh	0.0378	0.0045

Appendix M: A table showing the predictive loss across different loss functions for the latitude column.

N A table showing the predictive loss when experimenting on different loss function for the longitude column

Loss Function	MAE (degrees)	MSE (degrees)
Cosine Similarity	110.3147	12175.8649
Huber	0.2050	0.0524
Log Cosh	0.1944	0.0447
MSE	0.1505	0.0282
MSLE	4.3777	21.1968
MAE	0.1455	0.0287

Appendix N: A table showing the predictive loss across different loss functions for the longitude column.

O A table showing the predictive loss when experimenting on different loss function for the SOG column

Loss Function	MAE (knots)	MSE (knots)
Cosine Similarity	54.7028	3023.7500
Huber	0.4691	4.8010
Log Cosh	0.3972	4.7043
MSE	0.5029	4.1010
MSLE	0.4685	3.9703
MAE	0.3192	3.8145

Appendix O: A table showing the predictive loss across different loss functions for the SOG column.

P A Table showing the Structure of the Simplified RNN

Layer Type	Output Shape	Number of Parameters
Bidirectional GRU	(None, 7, 256)	103,680
Batch Normalisation	(None, 7, 256)	1,024
Bidirectional GRU	(None, 7, 256)	296,448
Batch Normalisation	(None, 7, 256)	1,024
Bidirectional GRU	(None, 7, 128)	123,648
Batch Normalisation	(None, 7, 256)	512
Bidirectional GRU	(None, 64)	31,104
Dense	(None, 16)	1,040
Dense	(None, 4)	68
Dense	(None, 1)	9

Appendix P: This shows the structure of the simplified model. It now includes batch normalisation layers and a reduced number of BiGRU layers. Generated using the summary function in Keras [29].

Q A table showing the Mean Predictive Window Size

Window Size	Mean MAE	Mean MSE
One	0.4511	2.3106
Two	0.3163	1.3844
Three	0.3303	1.3597
Four	0.3227	1.3213
Five	0.3726	1.5392
Six	0.2948	1.3534
Seven	0.3053	1.2892
Eight	0.3678	1.3702
Nine	0.2972	1.3495
Ten	0.2983	1.8840

Appendix Q: Shows the mean predictive error in MAE and MSE for each window size.