GEO-SPATIAL INFORMATION SCIENCE
2024, VOL. 27, NO. 1, 202–220
https://doi.org/10.1080/10095020.2022.2072241

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Extended linear regression model for vessel trajectory prediction with *a-priori* AIS information

Christiaan Neil Burger [a], Waldo Kleynhans [b] and Trienko Lups Grobler [a]

[a]Computer Science Division, Stellenbosch University, Stellenbosch, South Africa; [b]Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa

ABSTRACT

As maritime activities increase globally, there is a greater dependency on technology in monitoring, control, and surveillance of vessel activity. One of the most prominent systems for monitoring vessel activity is the Automatic Identification System (AIS). An increase in both vessels fitted with AIS transponders and satellite and terrestrial AIS receivers has resulted in a significant increase in AIS messages received globally. This resultant rich spatial and temporal data source related to vessel activity provides analysts with the ability to perform enhanced vessel movement analytics, of which a pertinent example is the improvement of vessel location predictions. In this paper, we propose a novel strategy for predicting future locations of vessels making use of historic AIS data. The proposed method uses a Linear Regression Model (LRM) and utilizes historic AIS movement data in the form of *a-priori* generated spatial maps of the course over ground (LRMAC). The LRMAC is an accurate low complexity first-order method that is easy to implement operationally and shows promising results in areas where there is a consistency in the directionality of historic vessel movement. In areas where the historic directionality of vessel movement is diverse, such as areas close to harbors and ports, the LRMAC defaults to the LRM. The proposed LRMAC method is compared to the Single-Point Neighbor Search (SPNS), which is also a first-order method and has a similar level of computational complexity, and for the use case of predicting tanker and cargo vessel trajectories up to 8 hours into the future, the LRMAC showed improved results both in terms of prediction accuracy and execution time.

ARTICLE HISTORY
Received 2 February 2021
Accepted 26 April 2022

KEYWORDS
Automatic Identification System (AIS) data; Linear Regression Model (LRM); trajectory mining; spatial map; historic data; trajectory prediction

1. Introduction

The world's oceans are of critical importance to humanity as they are key to fisheries, shipping, and the environment. From an economic perspective, it is estimated that 90% of all global goods and energy transportation are done by sea, with millions of people being dependent on maritime-related activities for their livelihood. As maritime activities increase globally, there is a greater dependency on technology in monitoring, control, and surveillance of vessel activities. One of the most prominent systems for monitoring vessel activity is the Automatic Identification System (AIS). AIS operates in the VHF band and transmits messages from vessels to other vessels, terrestrial shore stations, and satellites. Due to the global increase in vessels fitted with AIS transmitters and the proliferation of satellite and terrestrial receiving stations, there has been a significant increase in AIS messages received globally. This increased data volume makes it possible to track the real-time movement of vessels and opens the door for improving vessel location predictions via historic vessel movement patterns. Many algorithms have been developed in recent years to aid in improved vessel coordinate prediction. Methods range from simplistic models as done by Burger, Kleynhans, and Lups Grobler (2020) to machine learning (ML) models as done by Xin (2020).

Pallotta et al. (2014) presented a vessel prediction method based on Ornstein-Uhlenbeck stochastic processes, where the parameters of these processes are estimated from historical patterns in historic AIS data. The data is clustered into three types: vessels, waypoints, and routes. Route extraction is done using Traffic Route Extraction and Anomaly Detection from AIS Data (THREAD). The implementation details of THREAD can be found in Pallotta, Vespe, and Bryan (2013a, 2013b). The three types of clustered data aid in vessel prediction and empirical calculations. The maximum prediction time window of a vessel depends on the mean duration of the historically observed route.

Lee, Han, and Whang (2007) presented a trajectory clustering method, where similar trajectories are clustered together. The method works by employing a partition-and-group framework for clustering trajectories. It works by first grouping trajectories into a set of line segments, and similar segments are then grouped into clusters. The method consists of two phases, partitioning and grouping. In the first phase,

CONTACT Christiaan Neil Burger burgercnb@gmail.com

© 2022 Wuhan University. Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the minimum description length principle is used (Grünwald, Jae Myung, and Pitt 2005). In the second phase, a density-based line-segment clustering algorithm is used (Ester et al. 1996)based on a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN). The model's sensitivity on its parameters is improved in Jiashun (2012).

Rong, Teixeira, and Guedes Soares (2019) presented a probabilistic trajectory prediction model that describes the uncertainty in future spatial locations along a vessel's trajectory by using continuous probability distributions. A non-parametric Bayesian model based on a Gaussian Process (GP) is presented. The GP is used to describe the uncertainty of the vessel's motion, while the longitudinal uncertainty is derived from the dubiety of the vessel's acceleration. The parameters of the model are derived from historic AIS data. Cholesky decomposition is used to reduce the computational complexity of the algorithm. Prior probabilities are used to predict vessel coordinates in real-time.

Jaskolski (2017) implemented a Discrete Kalman Filter (DKF) (Kalman 1960) to predict future locations of vessels. The DKF, in the context of vessel coordinate prediction, constantly adjusts itself for an improved prediction as new observations are observed. It is assumed that a vessel fitted with an AIS sensor will not constantly send updates. The DKF consists of two sets of equations: prediction and measurement update equations used in prediction and parameter updates, respectively. Burger, Kleynhans, and Lups Grobler (2020), showed that there is no significant improvement in the prediction capability that can be achieved by using a DKF instead of a Linear Regression Model (LRM) to predict linear trajectories.

Xiao et al. (2020) conducted an extensive review of maritime knowledge mining and traffic forecasting technologies. The LRM is compared to several non-linear approaches. Three broad categories of non-linear algorithms are considered: machine learning approaches, knowledge-based approaches, and control theory assisted methods. The predictions range from long to short-term predictions. It is also shown that more complex methods are more accurate than the LRM but have a higher computational cost than the LRM.

Forti et al. (2020) made use of a deep learning (DL) neural network (NN) approach to predict trajectories of vessels. A sequence-to-sequence model that utilizes a Long Short-Term Memory (LSTM) encoded-decoder recurrent neural network (RNN) is proposed. Historic AIS data is used to train the LSTM model. The method aims to learn the predictive distribution of maritime traffic patterns using historic AIS data. Learning the predictive patterns enables the model to predict more accurately. It was shown that the model could predict more accurately than the Ornstein-Uhlenbeck process, given a time window of 20 observations.

Murray and Prasad Perera (2020) presented a novel dual linear autoencoder approach to predict a vessel's trajectory. The method predicts a future trajectory using historic AIS data. The method implements unsupervised learning for trajectory clustering and classification.

Xin (2020) presented a context-based trajectory prediction algorithm utilizing LSTM networks. Real-valued target trajectories are converted into discrete path sets. Distinctive patterns are clustered hierarchically using historical AIS data. Two models are compared, an RNN consisting of one LSTM and another RNN consisting of k LSTMs. In the RNN with k LSTMS, one LSTM is created for each distinct path. Yaun et al. (2019) used an LSTM model to reconstruct vessel trajectories. This complex method requires pre-processing and data clustering.

Dimitrios, Xidias, and Lekkas (2016) aim to accurately predict future geo-coordinates of a vessel by using artificial NNs (ANNs). Their model learns in real-time whilst predicting. It has a prediction time horizon of up to 15 minutes. Different types of model pre-processing and construction are implemented. Historic AIS data was used to train the ANN. Alizadeh, Asghar Alesheikh, and Sharif (2021) proposed three novel prediction methods based on historic AIS data. The first method proposed is a Point-based Similarity Search Prediction (PSSP), which was inspired by Wijaya and Nakamura (2013). The historical points are measured in terms of their spatial location, SOG, and COG. The second method proposed is called Trajectory-based Similarity Search Prediction (TSSP), where each recorded AIS trip is regarded as a trajectory. The PSSP is a point-based method, whereas the TSSP is a trajectory-based method. Finally, a trajectory-based similarity search prediction is proposed using an RNN LSTM (TSSPL). Alizadeh, Asghar Alesheikh, and Sharif (2021) point out that vessel movement is affected by external movements such as wind, waves, and sea currents. The PSSP and the TSSP are not able to account for these external factors. Another RNN LSTM model was, therefore, built to take this into account (i.e. TSSPL). The TSSPL has an additional input, a measure of similarity between trajectories (similar to what was done in Tang, Yin, and Shen 2019).

The Safety of Life at Sea (SOLAS) regulation V/19 states that passenger vessels and vessels larger than 300 Gross Tonnage must have AIS transmitters fitted. All cargo and tanker vessels should adhere to the SOLAS regulations. The regulations were adopted and are now regulated by the International Maritime Organization (IMO).[1] There are, however, some

drawbacks when working with AIS data. Vessels may deactivate their AIS transponders, and some vessels may not be fitted with transponders (Chen et al. 2019).

Other vessel tracking information systems include Long-Range Identification and Tracking (LRIT), Coastal radar, Satellite-borne Synthetic Aperture Radar (SAR), and optical satellites. Moreover, methods such as video tracking at ports exist where maritime video surveillance is present, as done in Chen et al. (2019) and Chen et al. (2020).

Finally, the method presented by Hexeberg, Flåten, and Brekke (2017) uses historic AIS data to predict future locations of vessels. The method is called Single-Point Neighbor Search (SPNS). The method does a close neighbor (CN) search by extracting historic observations within a certain radius of the current vessel's spatial location. Vessels in the CN set that do not adhere to prespecified SOG and COG range values are removed. Using the CN set, the median COG and SOG values are calculated. Using the median COG and SOG, the predicted longitude and latitude are calculated. The method predicts at constant distance intervals, where the SOG is used to calculate the time passed between two observations. The method can be confidently predicted with a time horizon of up to 15 minutes.

The aforementioned methods are complex in nature, being programmatically challenging to implement from first principles. The setup, initialization, and parameter estimation of these methods require fine-tuning to obtain optimal and accurate results. Burger, Kleynhans, and Lups Grobler (2020) showed that the complexity trade-off of using a DKF over an LRM resulted in no significant performance improvement in prediction accuracy, where the DKF was much more complex than the LRM. The DKF parameters are sensitive, and getting the initial set of parameters to be a reasonable estimate is crucial for the model's performance. Xiao et al. (2020) showed that the increase in performance gains comes with a significant increase in computational cost.

Therefore, this paper proposes a novel vessel location prediction method that extends the Linear Regression Model (LRM) proposed by Burger, Kleynhans, and Lups Grobler (2020). Moreover, this method uses historic AIS vessel information to improve prediction accuracy. It is easy to implement, and its initial parameters require little to no fine-tuning. The proposed algorithm also enables the LRM to predict non-linear trajectories. The *a-priori* information used includes a Spatial Map (SM) of historic cell counts, COG values, and COG standard deviations (SDs). The SMs the method uses are easy to generate and update, which is essential as data and movement patterns of vessels change over time. We refer to this method as the LRM with added COG (LRMAC) throughout the paper.

Due to the fact that the SPNS is also a simplistic first-order method (proposed by Hexeberg, Flåten, and Brekke 2017), the LRMAC will be compared to it. When comparing the models, an in-depth comparison of the model accuracies and time complexities will be made. Our tests will be implemented on cargo and tanker vessel data only. This paper is structured as follows: first, we discuss the dataset and the required processing. Second, we summarize the *a-priori* information the LRMAC requires. Third, we present the LRMAC and the comparison method, the SPNS. Fourth, we discuss the experimental design. Finally, we end the paper with our results and a conclusion.

2. Data

In this section, we will introduce the dataset that was used to test the performance of the proposed and benchmark algorithms. We will also be discussing the pre-processing steps that are necessary to ensure the proposed algorithm will run as intended.

An open-source dataset by Ray et al. (2019) was used. It consists of AIS messages recorded in the Celtic Sea, the North Atlantic Ocean, the English Channel, and the Bay of Biscay (France). In Figure 1, the spatial range of the data is depicted visually (also see the details in Table 1). The specific data characteristics are given in Table 2, and the observational period was over 6 months starting on 1 October 2015.

In Table 2, the dataset attributes and characteristics are introduced. The mathematical symbols that we use in this paper are presented in Table 3.

2.1. Pre-processing steps

Observations were removed from the dataset that did not adhere to the following criteria:

- Ship type within [70,89], where cargo vessel's type is in [70,79] and tankers in [80,89].
- SOG > 0.5kn, removing stationary observations, including stationary vessels experiencing drift due to currents and other natural phenomena due to being anchored.
- SOG < 60kn, observations with high speeds are likely outliers as cargo and tanker vessels move at relatively low speeds. If more than 60 kn is observed for cargo or tanker vessels, it is most likely due to technical errors in the recorded AIS data.

The remainder of the data were grouped according to vessel MMSI and sorted in ascending order according to the timestamp recorded by each observation. All trajectories with less than 20 observations or those that span less than 5 min in total were removed. The data spans over a period of 6 months, as denoted in Table 2, which implies that there will be more than one

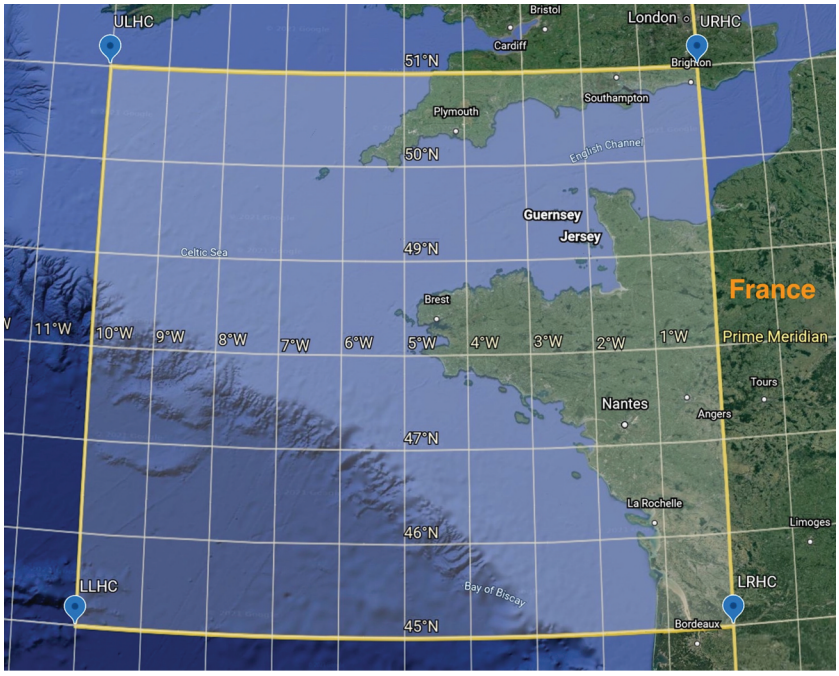

Figure 1. Data Spatial range – courtesy of Google Earth 2021.

Table 1. Spatial range details for Google Earth extract.

Name	Abbreviation	Longitude	Latitude
Upper Left-Hand Corner	ULHC	51	10
Upper Right-Hand Corner	URHC	51	0
Lower Right-Hand Corner	LRHC	45	0
Lower Left-Hand Corner	LLHC	45	10

Table 2. Dataset attributes.

Description	Measurement Unit	Attribute Range
MMSI		9-digit values
Latitude	DD.dddd	[−10.00 , 0.00]
Longitude	DD.dddd	[45.00 , 51.00]
Course Over Ground	Degrees	0–360
Speed Over Ground	Knots	0–110
Timestamp	UTC	[2015–10-01 00:00:00, 2016–03-31 23:59:59]
Ship type		[10, 99]

trajectory for a given vessel at different time periods and spatial locations. Pre-processing was done to remove any non-sensical data that may lead to incorrect SMs.

2.2. Vessel Interpolation for Spatial Maps (SMs)

After pre-processing, an augmented dataset was constructed by interpolating between observations for each unique vessel MMSI, creating less sparse trajectory observations per vessel. This augmented dataset is then used to construct physically realistic and usable spatial maps (SMs) of the dataset in question. The original dataset is too sparse (as is) for a meaningful and sensical SM to be created from it. The sparsity of the original data is due to several reasons. Firstly, vessels closer to receivers can share their position

Table 3. Variables and Symbol Descriptions.

Symbol	Meaning	Type	Dimensions
t	Time (DD-MM-YYYY HH:MM:SS)	Scalar	
Δk_t	Prediction time interval size (s)	Scalar	
ϕ	Latitude (DD.dddd)	Scalar	
λ	Longitude (DD.dddd)	Scalar	
ψ	Course Over Ground (°)	Scalar	
V	Speed Over Ground (kt)	Scalar	
V'	Speed Over Ground in Degrees	Scalar	
ω	Window Size	Scalar	
κ	SM cell length	Scalar	
ξ	Single SM cell	Scalar	
η	Neighborhood search size	Scalar	
$\boldsymbol{H}$	Neighborhood grid index values	Matrix	$(2\eta+1)\times(2\eta+1)$
$\boldsymbol{K}$	Vessel counts per cell SDM Matrix	Matrix	1250×1250
$\boldsymbol{\Psi}$	Course Over Ground SM Matrix (°)	Matrix	1250×1250
Σ	Course Over Ground SD SM Matrix (°)	Matrix	1250×1250
i_λ	Denotes the Longitude index of a SM matrix	Scalar	
i_ϕ	Denotes the Latitude index of a SM matrix	Scalar	

more often than vessels further away. Furthermore, vessels close to the shoreline travel at lower speeds than vessels further away. Naturally, more AIS messages will be recorded for slower vessels than faster vessels over the same distance traveled.

Observational interpolation was only performed in the following cases:

- The time difference between the two observations is no longer than six hours,
- The distance between the two observations is within 15 km, and
- The distance between observations is no smaller than the size of one grid cell. The grid cell size used was 0.88 km × 0.88 km. This specific constraint prevents the over-representation of a grid cell, only adding one observation to a cell if the interpolated trajectory passed through it.

We made use of linear interpolation models by using scikit-learn (Pedregosa et al. 2011). Longitude, latitude, and SOG were interpolated if the cases above were met. Gaussian filtering was also used to have smoother versions of the SMs.

The COG of the interpolated and recorded observations was calculated via

$$\psi_t = arctan\left(\frac{\phi_{t-1} - \phi_t}{\lambda_{t-1} - \lambda_t}\right) \tag{1}$$

We did not make use of the COG values present in the dataset. These values yielded inaccurate results, more accurate results were obtained by calculating the actual COG based on the longitude and latitude. Having a new augmented dataset with a better representation of the historical locations and trajectories, we can create representative SMs, essential to the algorithm we propose.

3. Spatial Maps (SMs)

Any two-dimensional grid that spans the earth's surface will be referred to as a spatial map. Each cell in the two-dimensional grid is associated with a specific range of longitudinal and latitudinal coordinates. In this paper, our SMs will be set up from the observations recorded in the dataset, which spans a latitude and longitude of $\phi \in [-10°, 0°]$ and $\lambda \in [45°, 55°]$, respectively.

In this paper, we use a square SM, meaning the SM's width and length have the same number of cells. The dimensions of the SMs depend on the range of the ϕ and λ. The dimensions of the SMs we constructed were 1250 × 1250 cells as shown in Table 3. The size of the SMs is 10° × 10° square degrees. Each cell's resolution, therefore, is equal to 0.008 × 0.008 square degrees. An extract of the upper left-hand corner of an artificially created SM is depicted in Figure 2. This figure denotes the number of recorded observations in a cell, within a longitude and latitude range based on the historic AIS data.

Latitude \ Longitude	0.008	0.016	0.024	...
0.000	0	0	2	0
0.008	0	2	5	0
0.016	1	2	5	0
0.024	2	5	1	0
⋮				

Figure 2. Spatial distribution map matrix extract example.

Haversine's formula for distance on a sphere is given by Equation (2). This formula is used to calculate the distance between two coordinates on a sphere.

$$d = 2r \cdot arcsin\sqrt{sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + cos(\phi_1)cos(\phi_2)sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \tag{2}$$

where,

- r represents the mean radius of the earth (≈ 6371 km).
- ϕ_1 and ϕ_2 represent the latitudinal coordinates of two observations, the latitudes of points one and two, respectively.
- λ_1 and λ_2 represent the longitudinal coordinates of two observations, the longitudes of points one and two, respectively.

It now follows from the Haversines formula that the area associated with each cell is roughly 0.89 × 0.89 km^2, where 0.89 km $\approx$ 0.48 nautical miles.

3.1. Vessel counts SDM (K)

The first type of SM that we use represents the number of observations recorded within each grid cell ξ. Each ξ in $\boldsymbol{K}$ records the number of observations that were historically within the longitude and latitude range of that specific cell. Cell counts will be higher in spatial areas where vessels are slower moving compared to faster-moving areas. Figure 3 shows the logarithmically scaled SDM of the vessel counts for each cell. The cell

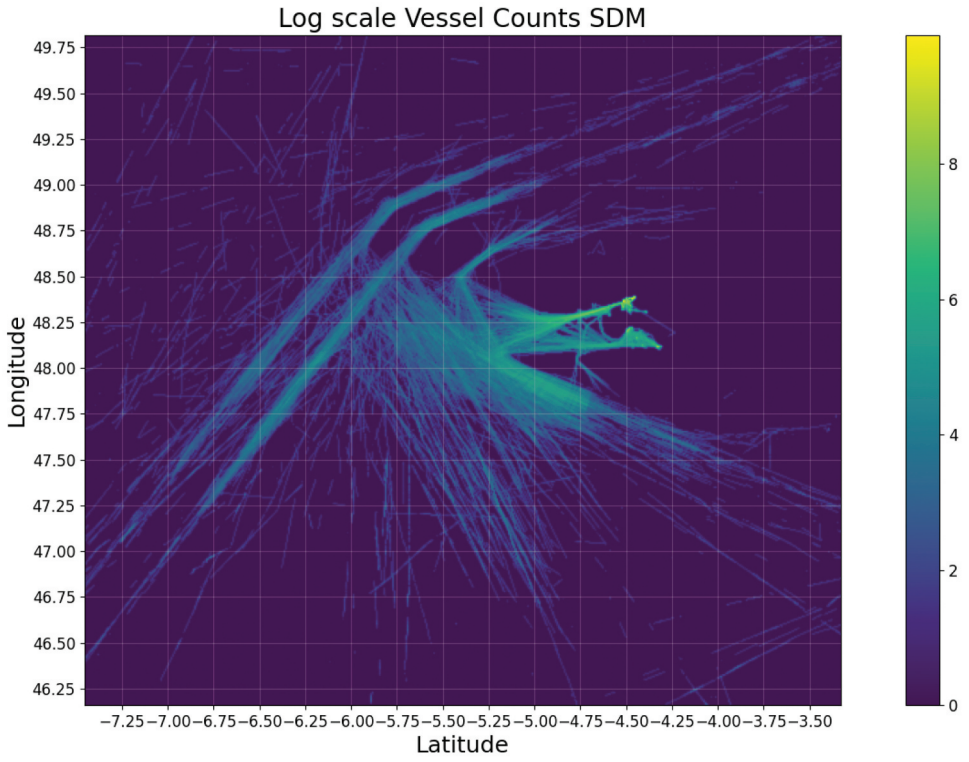

Figure 3. Log scale vessel counts SDM.

counts of the current vessel location and its neighbors are used in the prediction equations of the LRMAC, scaling the contribution of historic COG information.

3.2. The COG (Ψ) and COGSD (Σ) SM

The second SM that we will be using will be the course over ground SM (represented by $\mathbf{\Psi}$). Each cell of the $\mathbf{\Psi}$ represent the mean COG value that was recorded in that cell. COG is measured in degrees and will always be positive.

The third SM we used is the so-called COG standard deviation SM (represented by $\mathbf{\Sigma}$). The value of each of its cells was computed as follows:

$$\mathbf{\Sigma}_{i_\phi,i_\lambda} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\psi_{i_\phi,i_{\lambda_j}} - \mathbf{\Psi}_{i_\phi,i_\lambda}\right)^2} \tag{3}$$

where

- n, represents the number of observations observed in a cell as determined by $\boldsymbol{K}$.
- i_ϕ, i_λ, represents the index values associated with ϕ and λ respectively on the SM grid.
- $\mathbf{\Psi}_{i_\phi,i_\lambda}$, represents the mean COG at a specific index value on the SM grid.
- $\psi_{(i_\phi,i_\lambda)_j}$, represents the j^{th} COG value in the cell with index values i_ϕ, i_λ
- $\mathbf{\Sigma}_{i_\phi,i_\lambda}$, refers to the COG standard deviation at index value i_ϕ, i_λ.

Loosely speaking, the entries in $\mathbf{\Sigma}$ can be interpreted as how "confident" we ought to be in the corresponding entry in $\mathbf{\Psi}$. Higher cell values in $\mathbf{\Sigma}$ mean that historically there were many vessels traveling in different directions as the SD is higher.

In Figure 4, we can see a visual representation of $\mathbf{\Psi}$. Looking at the figure, we can see that cargo and tanker vessels move in a specific direction in certain geographic locations. Two distinct highways are clearly visible close to the center of Figure 4 (Grobler and Kleynhans 2019). A highway is a route that many vessels traverse. The aqua green highway is used by cargo and tankers to travel upward (North), while the blue highway is used to travel downward (South).

In Figure 5, we see a visual representation of $\mathbf{\Sigma}$, as calculated by Equation (3). The standard deviation of the areas that contain more traffic in different directions is larger than those containing less traffic (this is especially true for the areas surrounding harbors). Yellowish colors represent higher SD values. The highways mentioned, however, are not associated with high SD values. This implies that these highways are highly directional as shown in Figure 5.

4. The proposed algorithm

In this section, we present the proposed LRMAC algorithm. As mentioned before, this algorithm uses historic (*a-priori*) AIS data to predict the trajectories of vessels. The algorithm can predict non-

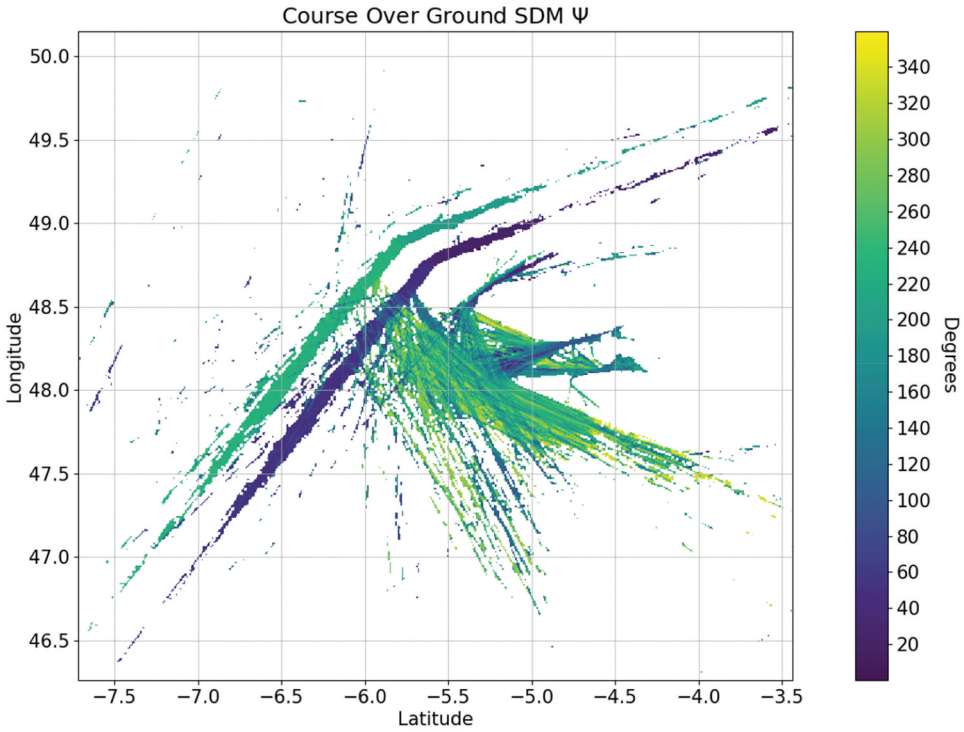

Figure 4. Course over ground SM.

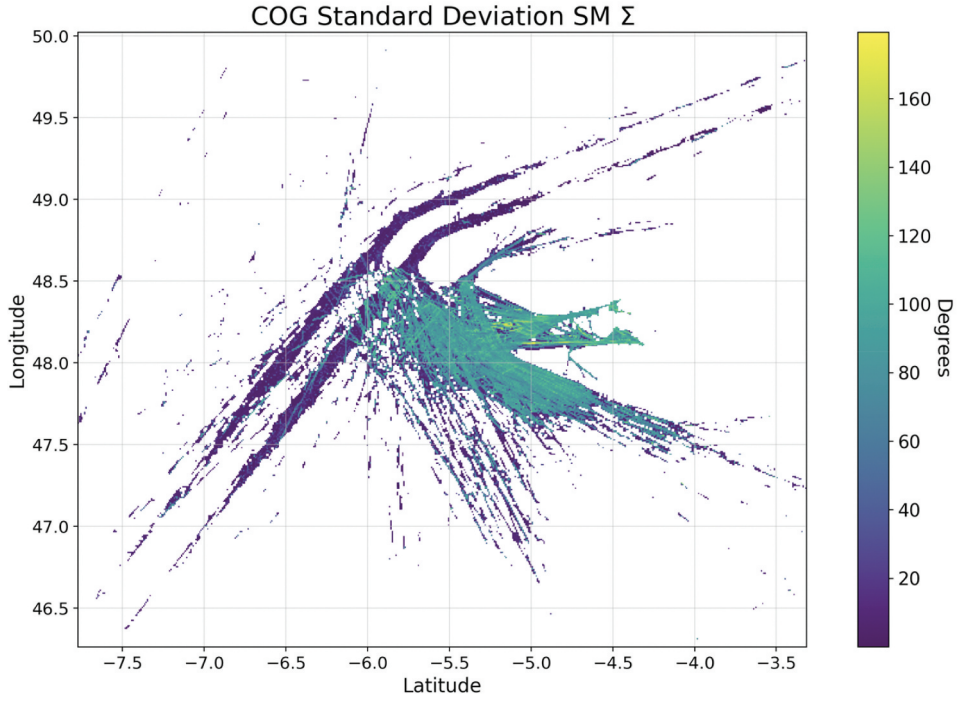

Figure 5. Course over ground standard deviation SM.

linear vessel trajectories. To be more specific, the LRMAC uses three SMs discussed in the previous section $\boldsymbol{K}, \boldsymbol{\Psi}$, and $\boldsymbol{\Sigma}$. We start this section by discussing the different unit conversions that the algorithm requires.

4.1. Speed Over Ground (SOG)

We converted SOG from knots to meters per second via Equation (4).

$$V_t^{'} = 0.514\dot{4} \cdot \mathrm{V}_{\mathrm{t}}^{''} \tag{4}$$

We then converted the SOG into degrees/s (°/s) using

$$\mathrm{V_t} = \frac{V_t^{'}}{l} \tag{5}$$

with

$$l = \frac{2\pi}{360} \times 6378000 = 111137 \text{ m} \tag{6}$$

The constant l, can be interpreted as the average number of meters that one degree of latitude and longitude span on Earth.

4.2. The Linear Regression Model (LRM)

The method presented in this paper (LRMAC) is an extension of the LRM presented by Burger, Kleynhans, and Lups Grobler (2020). We will summarize it below for the benefit of the reader. The LRM can only be predicted linearly with regular time intervals. The prediction interval that we used was 1 s, i.e. $\Delta k_t = 1s$. The LRM-based approach works by estimating V_t using a rolling window linear model.

One of the parameters that must be set is the prediction duration, i.e. the total number of hours one would like to predict into the future. The other parameter will be the step size of the predictions, measured in seconds between any two observations. The prediction interval can then be computed.

The LRM consists of two sets of equations: the predictor and measurement update equations. Let the vector $\boldsymbol{x}_t$ be defined as:

$$\boldsymbol{x}_t = [\lambda_t, \boldsymbol{\phi}_t]^T \tag{7}$$

The symbols $\boldsymbol{\phi}$ and λ are defined in Table 3. Subscript t refers to the observations at timestep t. If there is no observation at timestep $t, \boldsymbol{x}_t$ is assumed to be equal to the all-zero vector.

The prediction equations for the LRM are

$$\widehat{\boldsymbol{x}}_t^- = \widehat{\boldsymbol{x}}_{t-1} + \hat{V}_{\omega,t} \cdot \boldsymbol{\Lambda}_\mathrm{t} \tag{8}$$

$$\hat{V}_{\omega,t} = \Delta \hat{V}_{\omega,t-1} \cdot k_t + \hat{V}_{\omega,c_{t-1}} \tag{9}$$

$$\boldsymbol{\Lambda}_\mathrm{t} = \left[\cos(\psi_t)\ \sin(\psi_t)\right]^T \tag{10}$$

The variables in the equations above are defined below,

- $\widehat{\boldsymbol{x}}_t -$ denotes the vessel's predicted position vector using all observations up until timestep $t-1$.
- $\widehat{\boldsymbol{x}}_{t-1}$ denotes the vessel's updated estimated position vector using all observations up until timestep $t-1$.
- $\hat{V}_{\omega,t}$ denotes the vessels predicted SOG using all observations up until timestep $t-1$.
- $\hat{V}_{\omega,t-1}$ denotes the vessel's updated estimated gradient of our LRM using all observations up until timestep $t-1$.
- $\hat{V}_{\omega,c_{t-1}}$ denotes the updated estimated y-intercept of our LRM using all observations up until timestep $t-1$.
- ψ_t, denotes the COG of a vessel at timestep t, the COG remains the same until we observe a new COG value, i.e. getting an update from the vessel.
- ω, denotes the window size of our LRM.
- k_t, denotes the elapsed time in seconds at timestep t.

The measurement update equations are

$$\widehat{\boldsymbol{x}}_t = \widehat{\boldsymbol{x}}_t^- + \left(\boldsymbol{x}_t - \widehat{\boldsymbol{x}}_t^-\right) = \boldsymbol{x}_t \tag{11}$$

$$\nabla \hat{V}_{\omega,t} = \frac{\sum_{i=1_{n_t>\omega}(n_t-\omega)}^{n_t} \left(V_i - \bar{V}^\omega\right)\left(k_i - \bar{k}^\omega\right)}{\sum_{i=1_{n_t>\omega}(n_t-\omega)}^{n_t} \left(k_i - \bar{k}^\omega\right)} \tag{12}$$

$$\hat{V}_{\omega,c_t} = \bar{V}^\omega - \nabla \hat{V}_{\omega,t} \cdot \bar{k}^\omega \tag{13}$$

Moreover

$$1_{n_t>\omega} = \begin{cases} 1, \text{if } n_t > \omega \\ 0, \text{otherwise} \end{cases} \tag{14}$$

$$\bar{V}^\omega = \frac{1}{n_\omega} \sum_{i=1_{n_t>\omega}(n_t-\omega)}^{n_t} V_i \tag{15}$$

$$\bar{k}^\omega = \frac{1}{n_\omega} \sum_{i=1_{n_t>\omega}(n_t-\omega)}^{n_t} k_i \tag{16}$$

$$n_\omega = \begin{cases} n_t, \text{if } n_t < \omega \\ \omega, \text{otherwise} \end{cases} \tag{17}$$

The variables in the equations above are defined below:

- V_i denotes the i^{th} true SOG observation that was recorded for a particular vessel. We assumed $V_0 = 4\,\mathrm{m/s}$.
- k_i denotes the total time that has elapsed after having the i^{th} true observation.
- n_t denotes the total number of true observations recorded after $\Delta k_t \cdot t$ seconds, where in this paper $\Delta k_t = 1\mathrm{s}$. Δk_t refers to the prediction time interval between two subsequent observations.
- Note that subscript t is used as a time step index, while subscript i is used as an observational index.

The measurement update equation values are only updated once a new observation is recorded. If there are no new observations recorded, the algorithm will evaluate the prediction equations. During the period of no observations, it is assumed that the COG remains

unchanged. Moreover, the predicted position vector at $t-1$ is used to obtain a new estimate of the position vector at t.

4.3. LRM with a-priori COG information (LRMAC)

In this section, we will present our novel prediction method that is an extension of the LRM. The proposed method uses spatial maps as they can easily be loaded into memory, and their sizes are relatively small. The proposed method will allow for non-linear trajectory predictions of vessels. Note that the symbols used in this section should not be confused with those used in the previous section for the LRM.

The LRMAC algorithm no longer assumes a constant COG. The value of the COG is dynamically updated using *a-priori* information, using the three SM matrices $\boldsymbol{K}$, $\boldsymbol{\Psi}$ and $\boldsymbol{\Sigma}$. As the COG value is dynamically updated, the LRMAC can predict non-linear trajectories (i.e. the SOG in the longitudinal and latitudinal directions will change). It is assumed that the SOG will remain constant over the prediction period, where the constant SOG used during prediction is derived from the last ω observed SOG values of the vessel under consideration.

The predictor equations are modified for the LRMAC compared to the LRM. Whilst in prediction mode, the COG value is updated via *a-priori* information. The COG value computed in step t is only applied during step $t+1$.

Let $\widehat{\boldsymbol{x}}_t^-$ be the predicted position vector as in Equation (8). Let,

$$\boldsymbol{n}_{\hat{\phi},\hat{\lambda}} = \left[n_{\hat{\phi}}, n_{\hat{\lambda}}\right] \tag{18}$$

denote the SM index positions associated with (ϕ,λ). In other words, with $\left[n_{\hat{\phi}}, n_{\hat{\lambda}}\right]$ we can extract the elements in $\boldsymbol{K}$, $\boldsymbol{\Psi}$ and $\boldsymbol{\Sigma}$ associated with (ϕ, λ) by making use of matrix subscripting. Let us now construct the following index matrix H:

$$H = \begin{bmatrix} \boldsymbol{n}_{\hat{\phi}-\eta\cdot\kappa,\hat{\lambda}-\eta\cdot\kappa} & \cdots & \boldsymbol{n}_{\hat{\phi}-\eta\cdot\kappa,\hat{\lambda}} & \cdots & \boldsymbol{n}_{\hat{\phi}-\eta\cdot\kappa,\hat{\lambda}+\eta\cdot\kappa} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \boldsymbol{n}_{\hat{\phi}\cdot\kappa,\hat{\lambda}-\eta\cdot\kappa} & \cdots & \boldsymbol{n}_{\hat{\phi},\hat{\lambda}} & \cdots & \boldsymbol{n}_{\hat{\phi},\hat{\lambda}+\eta\cdot\kappa} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \boldsymbol{n}_{\hat{\phi}+\eta\cdot\kappa,\hat{\lambda}-\eta\cdot\kappa} & \ldots & \boldsymbol{n}_{\hat{\phi}-\eta\cdot\kappa,\hat{\lambda}} & \ldots & \boldsymbol{n}_{\hat{\phi}+\eta\cdot\kappa,\hat{\lambda}+\eta\cdot\kappa} \end{bmatrix} \tag{19}$$

where

- η in Equation (19) denotes the neighborhood parameter, and
- κ represents the width and length of an SM cell (see Table 3 and Figure 2).

The index matrix H is used to select a specific sub-grid /matrix from K and Σ.

Let the *a-priori* cell counts, of the area surrounding $\widehat{x}_t^-$, therefore, be denoted by

$$\boldsymbol{K}_H, \text{where } \boldsymbol{K}_H \subset \boldsymbol{K}$$

The size of the aforementioned area is determined by η. Moreover, let the *a-priori* average COG value associated with $\widehat{\boldsymbol{x}}_t^-$, be denoted by

$$\boldsymbol{\Psi}_{n_{\hat{\phi},\hat{\lambda}}}, \text{where } \boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}} \in \boldsymbol{\Psi}$$

Finally, let the COG SD associated with $\widehat{\boldsymbol{x}}_t^-$, be defined as,

$$\boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}, \text{where } \boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}} \in \boldsymbol{\Sigma}$$

4.3.1. Updating the COG using a-priori information

We first need to calculate the confidence we have in the *a-priori* COG value $\boldsymbol{\Psi}_{n_{\hat{\phi},\hat{\lambda}}}$. This confidence measure ρ allows ranges between $[0, 1]$, and it is used to scale the contribution of the *a-priori* COG information. It is determined by the number of observations in the neighborhood, as shown by Equation (20). If the *a-priori* grid count for our current cell is high compared to the other cell counts in the neighborhood, we can more confidently say that the predicted location is in an area where historically many vessels have traveled before. If the observed cell count for the current position is lower than in the surrounding cells, we will give less confidence in the *a-priori* COG update and rather assign more confidence in the LRM with the scaling factor. We can achieve this by calculating the confidence factor ρ:

$$\rho = 1_{\boldsymbol{\Psi}} \cdot \frac{\boldsymbol{K}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}}{\max(\boldsymbol{K}_{\boldsymbol{H}})} \tag{20}$$

where

- ρ denotes the confidence (scaling factor) that we have in our prediction as determined by the current position of the vessel.
- $\boldsymbol{K}_{n\frown}$ denotes the cell count value associated with $\left(\hat{\phi},\hat{\lambda}\right)$. The index $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ is used to extract it from $\boldsymbol{K}$.
- $1_{\boldsymbol{\Psi}}$ denotes the indicator function that sets ρ to zero. The indicator function enforces the restrictions that we impose on whether the COG should be updated or not.

ρ determines the total weight the *a-priori* COG information should have in the LRMAC. If ρ is close to one, we can be confident in the *a-priori* value $\boldsymbol{\Psi}_{n_{\hat{\phi},\hat{\lambda}}}$, and if it is close to zero, we are less confident, and it should have less of a contribution in our predictions, as seen in Equation (22). Two factors influence ρ. First,

if a few vessels have traversed the cell associated with x_t relative to its neighboring cells, $\frac{K_{n_{\hat{\phi},\hat{\lambda}}}}{\max(K_H)}$ would be relatively close to zero, indicating that the *a-priori* COG information should have less impact in updating Ψ_{t+1}. The same is true for the opposite: values closer to one would have a more significant contribution to updating Ψ_{t+1}. The second factor is determined by the value of the indicator function in Equation (21).

$$1_{\Psi} = \begin{cases} 0, \ \Psi_{n_{\hat{\phi},\hat{\lambda}}} \notin \Re \\ 0, \Sigma_{n_{\hat{\phi},\hat{\lambda}}} \notin \Re \\ 0, \Sigma_{n_{\hat{\phi},\hat{\lambda}}} > 10^{\circ} \\ 0, max(K_H) = 01, otherwise \end{cases} \quad (21)$$

This equation will evaluate to zero if,

- Ψ or Σ contains no information at index $n_{\hat{\phi},\hat{\lambda}}$. This implies that there is no *a-priori* information available for us to make use of.
- The square root of the COG SD at $n_{\hat{\phi},\hat{\lambda}}$ is larger than 10° i.e. standard deviation. This implies that many vessels have traversed through the cell associated with $\hat{x}_t$, all going in different directions. This implies that the *a-priori* COG value is unreliable.

The COG value can now be updated as follows:

$$\hat{\psi}_{t+1} = (1-\rho)\psi_t + \rho\Psi_{n_{\hat{\phi},\hat{\lambda}}} \quad (22)$$

where

- $(1-\rho)$ indicates the role that the previous observed COG should have in the COG update.
- ψ_t denotes the previously observed or predicted COG.
- ρ denotes the scaling factor that the historic *a-priori* COG information should have.
- $\Psi_{n_{\hat{\phi},\hat{\lambda}}}$, denotes the a-priori COG value at index $n_{\hat{\phi},\hat{\lambda}}$.

If our confidence in the *a-priori* COG value is high, we weigh it accordingly. If it is low, we rather put more trust in the COG value at time step t (weighted by ρ). Note the estimated COG will replace the COG value that is used in Equation (10).

4.3.2. A flow diagram representation of the LRMAC

In Figure 6, a flowchart of the LRMAC methodology is depicted. The flowchart depicts the LRM and how *a-priori* course information is added to extend the LRM into the LRMAC. Initializations are in green, functions are in blue, the predicted location is in gray, and parameter extracts are denoted in orange. It is assumed that all pre-processing has already been applied to the dataset.

The first step is to construct the SMs from the dataset containing all historic AIS data and initialize all parameters. The last ω recorded observations are used as additional input parameters in the LRM, specifically for the measurement update equations. The LRM consists of two sets of equations, measurement update and predictor equations. The measurement equations are used to update the predictor parameters and the predictor equations to predict the next set of coordinates.

The algorithm starts at the indicated red dot in the flowchart. The LRM is used to estimate the longitude and latitude at the next time step. All predictions are made at regular spaced time intervals Δk_t. The LRM assumes a constant SOG and COG, based on the last ω observations.

Given the predicted longitude and latitude, we extend the LRM into the LRMAC by dropping the constant COG assumption. The COG will now be updated based on *a-priori* COG information at the

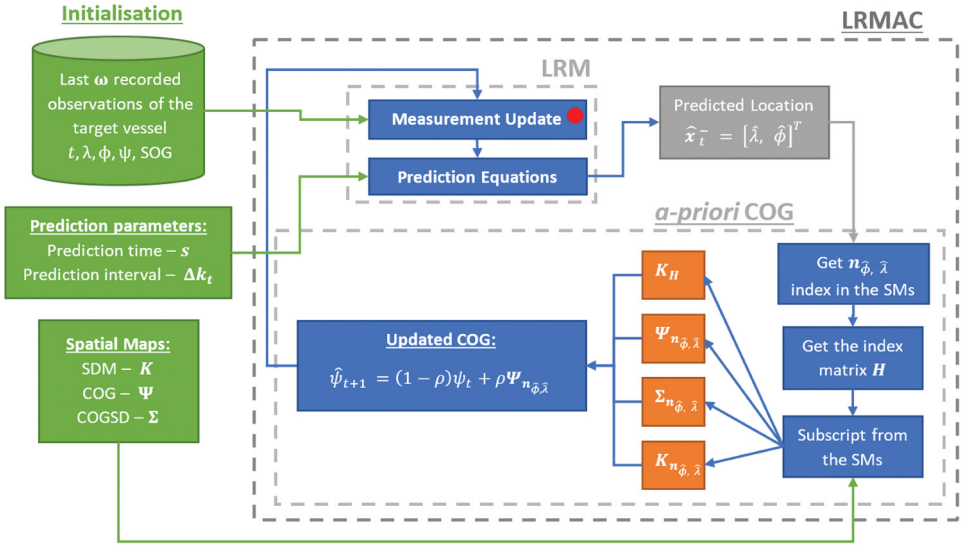

Figure 6. LRMAC flowchart.

location predicted by the LRM. The extension of the LRM allows the predictions to follow historic movement trends in the SMs, allowing for non-linear trajectory prediction.

Given the predicted location $\hat{x}_t$, the corresponding index $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ in the SMs can be calculated, and the neighboring indexes matrix $\boldsymbol{H}$. Using $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ and H, the *a-priori* values: $\boldsymbol{K_H}$, $\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$, $\boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$, and $\boldsymbol{K}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$ is extracted and used in Equations (20) and (21) to calculate the updated COG value $\hat{\psi}_{t+1}$. The updated COG is used in the next iteration by updating the measurement equations of the LRM, which in turn updates the predictor equations. This allows for a COG that updates dynamically based on historic AIS data.

All other parameters, including the COG, will be updated once new observations are received from the vessel, updating all the measurement equations.

4.4. Single-Point Neighbor Search (SPNS)

We will now be summarizing the method presented by Hexeberg, Flåten, and Brekke (2017) called Single-Point Neighbor Search (SPNS). We will be comparing the LRMAC to the SPNS, as it was one of the most similar methods we found. We will compare the prediction accuracy and the time complexity of each. Note, do not confuse the variables defined in this section with the variables used in the rest of the paper.

Let

$$\boldsymbol{X} = [\boldsymbol{X}_1\, \boldsymbol{X}_2\, \ldots\, \boldsymbol{X}_M]^T \tag{23}$$

be defined as a matrix with all the historic AIS data observations, where M indicates the number of AIS messages recorded.

Let

$$\boldsymbol{X}_i = \left[MMSI_i\, t_i\, \boldsymbol{p}_i^T\, \chi_i\, v_i\right] \tag{24}$$

be defined as a vector, where $i \in \{1, 2, \ldots, M\}$. The elements in X_i refer to the MMSI, timestamp, position vector, COG, and SOG of timestep i. The position vector $\boldsymbol{p}_i^T$ denotes the spatial location of a vessel, where $\boldsymbol{p}_i = [\lambda_i, \boldsymbol{\phi}_i]^T$ denotes the longitude and latitude at message i.

A predicted trajectory consists out of K^s predicted positions, which are instants of time. At every iteration k, a prediction is made, where $k \in 1, \ldots, K^s$. The predicted state is divided into an *a-priori* state $\widehat{X}_i^{k-}$ and *a-posteriori* state $\widehat{X}_i^{k+}$. The states are denoted as:

$$\widehat{\boldsymbol{X}}_i^{k-} = \left[MMSI_i\, \hat{t}^k \widehat{\boldsymbol{p}}^k \hat{\chi}^{k-} \hat{v}^{k-}\right] \tag{25}$$

and

$$\widehat{\boldsymbol{X}}_i^{k+} = \left[MMSI_i\, \hat{t}^k \widehat{\boldsymbol{p}}^k \hat{\chi}^{k+} \hat{v}^{k+}\right] \tag{26}$$

The only difference between the two equations above, is the COG and SOG. The predicted $\hat{\chi}^{k-}$ and $\hat{v}^{k-}$ at iteration k in the *a-priori* state, represent the predicted COG and SOG between the previous position $\widehat{\boldsymbol{p}}^{k-1}$ and the current position $\widehat{\boldsymbol{p}}^k$. The *a-posteriori* predicted COG and SOG at iteration k is the difference between the current position $\widehat{\boldsymbol{p}}^k$ and the next position $\widehat{\boldsymbol{p}}^{k+1}$.

The SPNS makes use of a close neighbor search, where a set radius is defined as a parameter and observations within the radius is queried, given that the observations adhere to a set of predefined constraints.

Let the close neighbors (CNs) at prediction step k be defined as:

$$C^k = \{\boldsymbol{X_i} | d\left(\widehat{\boldsymbol{p}}^{\boldsymbol{k}}, \boldsymbol{p}_i\right) \leq r_c, \chi_i \in S, \boldsymbol{X}_i \in \boldsymbol{X}\} \tag{27}$$

where

- $d\left(\widehat{\boldsymbol{p}}^{\boldsymbol{k}}, \boldsymbol{p}_i\right)$ is defined as the Haversine distance between the longitude and latitude in $\widehat{\boldsymbol{p}}^{\boldsymbol{k}}$ and $\boldsymbol{p}_i$. The Haversine distance is defined in Equation (2).
- r_c, is defined as the search radius in meters, to search for all the CNs within r_c of the current position. The value of r_c is a predefined parameter.
- S is defined as the interval of course angles. Only observations within the course angle interval will be included in the CN set.

Let the course angles S be defined by:

$$S = \left[\hat{\chi}^{k-} - \Delta\chi, \hat{\chi}^{k-} + \Delta\chi\right] \tag{28}$$

where $\Delta\chi > 0$, is the maximum course angle deviation, which is a predefined parameter.

The above pre-processing steps of the distance and COG deviation will filter out all the observations that we do not need for the remaining steps. All the CNs are extracted from historic AIS data. Let every state that belongs to the set of CNs at the prediction step k be denoted as $\boldsymbol{X}_i \in C^k$, where $\boldsymbol{X}_c^k = \left[MMSI_c^k\ t_c^k\ \chi_c^k\ v_c^k\right]$ and $c \in \{1, \ldots, C\}$. C is the number of CNs at k.

The predicted trajectory at a state X_i is defined as,

$$\widehat{\boldsymbol{T}}_{\boldsymbol{i}} = \left\{\left[\widehat{\boldsymbol{p}}^1 \hat{t}^1\right], \ldots, \left[\widehat{\boldsymbol{p}}^K \hat{t}^{K^s}\right]\right\} \tag{29}$$

Let the true trajectory, T_i, given state X_i, be defined as

$$\boldsymbol{T}_i = \left\{\left[p^1 t^1\right], \ldots, \left[p^L t^L\right]\right\} \tag{30}$$

L denotes the number of AIS states recorded. K^s and L are not necessarily equal as several prediction steps can be made between two subsequent AIS messages. The first elements in both $\widehat{\boldsymbol{T}}_i$ and $\boldsymbol{T}_i$ are equal, as it is the starting point given by state $\boldsymbol{X}_i$.

4.4.1 SPNS prediction

A new parameter is introduced, Δl, which denotes the step length from the current observation to the next predicted observation in meters. Δl decides how far the next position should be propagated. Let the predicted position be denoted by

$$\hat{\boldsymbol{p}}^{k+1} = \boldsymbol{p}^{k} + \Delta l \cdot \left[\sin(\hat{\chi}^{k+}) \cdot f\left(\hat{\phi}^{k}\right) \cos(\hat{\chi}^{k+}) \cdot g\left(\hat{\phi}^{k}\right)\right]^{T} \quad (31)$$

$f\left(\hat{\phi}^{k}\right)$ and $g\left(\hat{\phi}^{k}\right)$ are functions of the current latitude $\hat{\phi}^{k}$, which transforms from meters to degrees longitude and latitude, respectively. The step length, Δl, reflects the curvature of the sea lanes ahead.

4.4.1.1 Course prediction.

The COG value $\hat{\chi}^{k+}$ is used when calculating the predicted position $\hat{\boldsymbol{p}}^{k+1}$. $\hat{\chi}^{k+}$ is the *a-priori* course calculated from the extracted CNs at position $\boldsymbol{p}^{k}$. Note that the course is periodic in $[0°, 360°]$; special care must be taken when calculating the CN set's mean COG. The mean COG is calculated as follows:

$$\bar{\chi}_c = \begin{cases} \tan^{-1}\left(\frac{\bar{s}}{\bar{c}}\right) if\, \bar{s} > 0, \bar{c} > 0 \\ \tan^{-1}\left(\frac{\bar{s}}{\bar{c}}\right) + 180° if\ \bar{c} < 0 \\ \tan^{-1}\left(\frac{\bar{s}}{\bar{c}}\right) + 360° if\ \bar{s} \langle 0, \bar{c} \rangle 0 \end{cases} \quad (32)$$

where

$$\bar{s} = \frac{1}{C}\sum_{c=1}^{C} \sin(\chi_c) \quad (33)$$

$$\bar{c} = \frac{1}{C}\sum_{c=1}^{C} \cos(\chi_c) \quad (34)$$

A constant velocity model is used whenever $C^{k} \notin \mathbb{R}$ as done by Hexeberg, Flåten, and Brekke (2017). The median course $\tilde{\chi}_c$ can be calculated by calculating $\tilde{s}$ and $\tilde{c}$. Hexeberg, Flåten, and Brekke (2017) recommended to use the median course for non-linear trajectories.

4.4.1.2 Speed prediction.

The median $\tilde{v}_c$ of the CNs is used to calculate the predicted speed. Note that the speed prediction is only used in the time update equation shown in Algorithm 1. The predicted speed $\tilde{v}_c$ is used to calculate the time passed between the current observation and the predicted observation, where the time passed is denoted by $\frac{\Delta l}{\hat{v}^{k+}}$. The time update equation is defined as:

$$\hat{t}^{k+1} = \hat{t}^{k} + \frac{\Delta l}{\hat{v}^{k+}} \quad (35)$$

where

- $\hat{t}^{k}$ denotes the current time,
- Δl denotes the distance between the current and predicted observation, and
- $\hat{v}^{k+} = \tilde{v}_c$, given set C^{k} at k.

Table 4. Curved trajectory decision parameters.

Decision Parameter	Value	Explanation
r_c	50 m	Search radius of for the CNs
Δl	$2r_c$	Prediction step length [m]
$\Delta\chi$	25°	Maximum course deviation
$\hat{\chi}_i^{k+}$	$\tilde{\chi}_c$	Course prediction used at every iteration k
$\hat{v}_i^{k+}$	$\tilde{v}_c$	Speed prediction used at every iteration k

The calculation of the predicted time allows the SPNS to have regular spaced distance intervals of length Δl. The algorithm for the SPNS is presented below.

Algorithm 1 Single Point Neighbor Search Prediction

1: X_i given
- The state we predict from

2: Set decision parameters
   (a) Δl
   (b) r_c
   (c) ΔX
   (d) K^s
- Step length [m]
- Search radius [m]
- Maximum course angle deviation [deg]
- Number of prediction steps

3: Set $\hat{X}_i^{k-} = X_i$
4: **for** $k = 1$ to K **do**
5: Find all CNs X_c^k around $\hat{X}_i^{k-}$
6: Calculate $\hat{X}_i^{k+}$ *by* :
   (a) Calculating $\hat{\chi}^{k+}$ based on X_c^k
   (b) Calculating $\hat{v}^{k+}$ based on X_c^k
7: Calculate the next predicted position at its predicted point in time:
   (a) Calculate $\hat{\boldsymbol{p}}^{k+1}$ according to Equation (31)
   (b) Calculate $\hat{t}^{k+1} = \hat{t}^{k} + \frac{\Delta l}{\hat{v}^{k+}}$
8: Set $\hat{X}_i^{(k+1)-} = \left[MMSI_i\ \hat{t}^{k+1} \hat{\boldsymbol{p}}^{k+1} \hat{\chi}^{k+} \hat{v}^{k+}\right]$
9: **end for**

In Table 4, the set of hyperparameters that are used for curved trajectories is shown. These decision parameters are identical to the ones presented by Hexeberg, Flåten, and Brekke (2017) and were used to compare the SPNS to the LRMAC.

5. Experimental design

This section will discuss how we set up our experiments to get the optimal set of parameters for the proposed LRMAC model. We will also be discussing how we will compare the LRMAC to the SPNS. We will be comparing the LRM and LRMAC to show the increase in prediction accuracy of the LRMAC over the LRM and then compare the LRMAC to the SPNS model comparing prediction accuracy and time complexity. We will be discussing the vessels used, how we extracted their trajectories, the subsampling approach that we made use of, and how we determined the performance of the two approaches.

To compare the LRMAC and SPNS, 40 unique vessel trajectories were extracted at random from the dataset published by Ray et al. (2019) after having been pre-processed. The pre-processing removed stationary observations and vessels with too few observations, as mentioned earlier in the paper. The extracted vessels are a mix of cargo and tanker vessels. The extracted

trajectories of the vessels were all from the high-density areas depicted in Figure 3. We restricted our use case to this example to show the performance of the LRM, LRMAC, and SPNS on curved trajectories. The LRMAC algorithm can be applied to various use cases and will perform well on any SM containing clear regions exhibiting path directionality.

5.1 SPNS vessel query setup

Since the SPNS algorithm requires querying historic observations within a specified radius, all the data was loaded into a PostgreSQL database.[2] An extension was added to PostgreSQL called PostGIS,[3] which allows for improved spatial queries with a datatype called geometry. The PostGIS plugin uses a unique kind of indexing. Querying observations in PostgreSQL with PostGIS allow that vessels within a given radius from the search point can be extracted into C^k.

5.2 Trajectory subsampling method

In order to have a large test set of trajectories to identify the ideal parameters for the LRMAC and compare it to the SPNS, we created a method that would subsample trajectories. We are effectively creating multiple trajectories from one observed trajectory. A given trajectory $\boldsymbol{T}_i$ will be subsampled into different time subsets given a prediction length h. Each time a subset's starting observation will differ by 1 hour compared to the subset that precedes or follows it. We refer to this hour difference as a stride length s, where s defines the starting point of the stride hour from $\boldsymbol{T}_i$. Stride values are measured in hours. This method allows us to extract multiple sub trajectories $T_{s,h}$ from $\boldsymbol{T}_i$. Let the number of sub trajectories that can be created with a prediction length of h from $\boldsymbol{T}_i$ be denoted by

$$\#T_h = \lfloor max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{hour} - h + 1 \tag{36}$$

and let the total number of sub trajectories from $\boldsymbol{T}_i$ with different starting positions s be denoted by,

$$\#T_{s,h} = \sum_{h=1}^{\lfloor max(\boldsymbol{t}_{T_i}) \rfloor_{hour}} \#T_h \tag{37}$$

where

- $\lfloor max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{hour}$ denotes the closest floored hour to the maximum observed time in T_i.
- h refers to the prediction length, where $h \in \{1, 2, \ldots, \lfloor max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{hour}$, and
- s denotes the stride starting position measured in hours,

$$s \in \{0, 1, \ldots, \lfloor max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{hour} - h\}$$

In Figure 7, an example is shown of the trajectory subsampling and the number of subsets we can create given an observed trajectory between 6 and 7 hours. Using Equation (37), the total number of sub trajectories that can be created from this trajectory is 21.

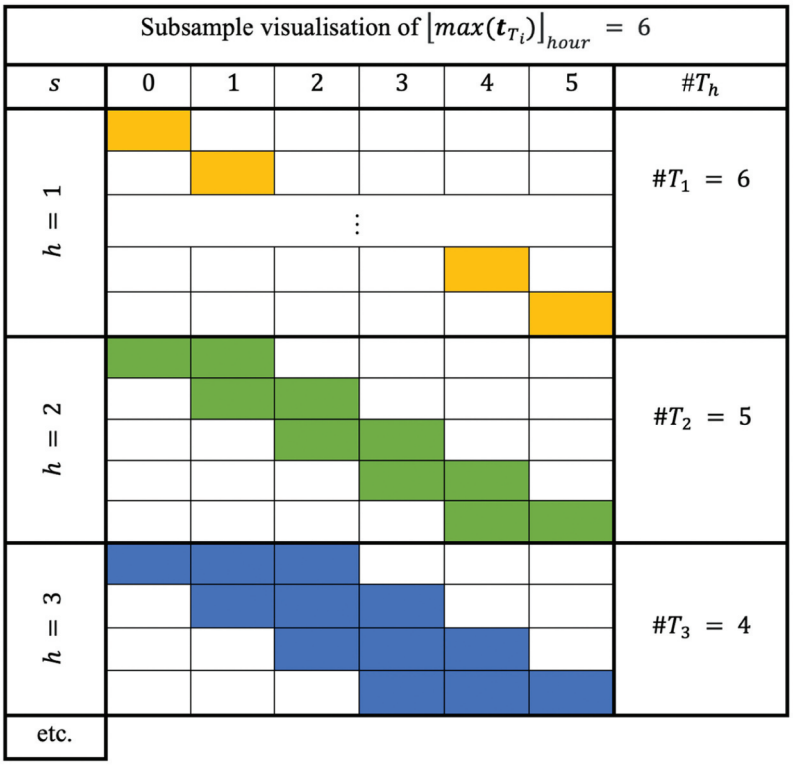

Figure 7. Subsample visualization.

This method of sub-trajectory sampling allows us to have multiple trajectories to compare the prediction and time performance between the LRMAC and the SPNS model. The prediction performance is determined by the Haversine distance between the expected spatial location at time t compared to the predicted location at time $\hat{t}$. The time performance is computed by the time the methods take from execution until the predicted trajectory of length h has been calculated.

First, we will run the LRMAC method on all sub-trajectories with different sets of parameters η and ω to determine the optimal set of parameters that results in the smallest median Haversine distance on average. The LRMAC with its optimal parameters and SM densities will be compared to the LRM, and SPNS with the parameters defined by Hexeberg, Flåten, and Brekke (2017). The SM densities were not optimized for these tests, the cell sizes remained constant, and the number of cells fixed at 1250×1250. We will also be testing the speed performance between the two algorithms, measuring the execution times.

5.3 Parameter estimation of LRMAC

To obtain the optimal set of hyperparameters for the LRMAC, the LRMAC was tested extensively on different combinations of η and ω. The combination of η and ω that gave the lowest overall Haversine distance error will be chosen as the ideal set of hyperparameters. The considered parameter values were: $\eta = [1, 2, 3, 4]$ and $\omega = [1, 3, 5, 7, 9]$, resulting in 20 combinations.

The parameters that yielded the best results were: $\omega = 3$ and $\eta = 1$. The LRMAC was tested on all sub trajectories. The combination of ω and η with the smallest average median Haversine distance over all sub-trajectories and timeframes was chosen as the ideal set of hyperparameters. Figure 8 compares the average median Haversine distances, and we can see that there is no significant performance increase after the combination $\omega = 3$ and $\eta = 1$. The chosen hyperparameters have a lower complexity as the ω and η are small, compared to the other parameters with similar performance. Smaller values of ω means that our function is more flexible to incorporate new information, as there are fewer elements to consider. Large values of ω will mean that our algorithm is less flexible, using more observed observations from the current trajectory to make a prediction. Larger values of η means more historic SM cells to include when making a prediction, as the neighborhood size is larger.

5.4 SPNS prediction adjustment

Since the SPNS predicts in constant distance intervals Δl instead of constant time intervals as the LRMAC does, we had to adjust the final prediction of the SPNS to allow for an exact comparison at h (prediction horizon). To get the predicted spatial location of the SPNS after h has passed, we let the SPNS predict until $\hat{t}^{k+1} > h$. We calculate the time that has passed between the last two predicted observations $\widehat{\boldsymbol{X}}_i^{(k)-}$ and $\widehat{\boldsymbol{X}}_i^{(k+1)-}$, where time is indicated by $\hat{t}^k$ and $\hat{t}^{k+1}$, and where, $\hat{t}^k < h < \hat{t}^{k+1}$., let he total time passed be denoted by $\Delta t_{|\hat{y}|}$. The Haversine distance between $\widehat{\boldsymbol{X}}_i^{(k)-}$ and $\widehat{\boldsymbol{X}}_i^{(k+1)-}$ is calculated $\Delta l_{\left|\widehat{X}_i^{(k)-} - \widehat{X}_i^{(k+1)-}\right|}$. The SPNS will be rerun from $\widehat{X}_i^{(k)-}$ with the same set of parameters and only changing the prediction step length $\Delta l = \Delta l_{\left|\widehat{X}_i^{(k)-} - \widehat{X}_i^{(k+1)-}\right|}$. This will give us the predicted spatial location at h, as $\hat{t}^{k+1} = h$ determined by Equation (35).

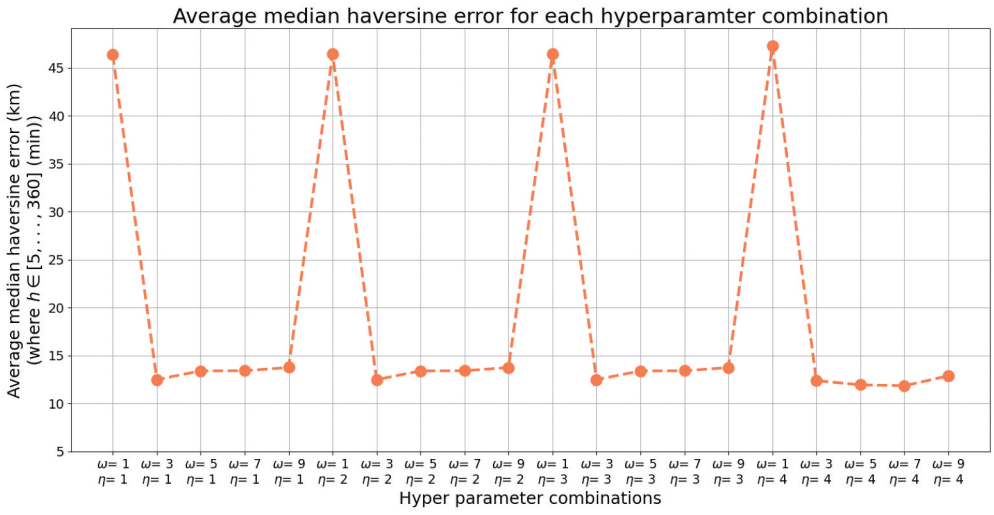

Figure 8. Average median haversine error for each hyperparameter combination.

5.5 Prediction performance comparison algorithm

The algorithm below shows how the errors were calculated for both the LRMAC and SPNS on each sub trajectory. We used the median error results for each predicted timeframe h to compare the performance between the two algorithms.

Algorithm 2 Prediction Performance Comparison

1: **Set different parameters**:
   (a) $\omega = 3$
   (b) $\eta = 1$
   (c) $\in_{\text{LRMAC}} = \left[\in_{\text{LRMAC}_1}, \ldots, \in_{LRMAC_{max(h)}}\right]$
   (d) $\in_{\text{SPNS}} = \left[\in_{SPNS_1}, \ldots, \in_{SPNS_{max(h)}}\right]$
   - LRMAC Jagged array for each h
   - SPNS Jagged array for each h
2: **for** T_i *in* T **do**
3: Get $T_{s,h}$ which is a set of sub trajectories from T_i
4: **for** h in $1, 2, \ldots, \lfloor max(t_{T_i})\rfloor_{hour}$ **do**
5: **for** s in $\{0, 1, \ldots, \lfloor max(t_{T_i})\rfloor_{hour} - h\}$ **do**
6: $T_{test} = T_{s,h}[0:3]$
   - Initial observations for the algorithms
7: $\hat{\lambda}_{LRMAC}, \hat{\phi}_{LRMAC} = LRMAC(T_{test}, \eta, \omega, h)$
   - LRMAC, given the parameters and h
8: $\hat{\lambda}_{SPNS}, \hat{\phi}_{SPNS} = SPNS(T_{test}, h)$
   - SPNS, given the parameters and h
9: **Calculate the haversine distance error**:
   $\in_{\text{LRMAC}}$ append ***havdist***$(T_{s,h}, \hat{\lambda}_{LRMAC}, \hat{\phi}_{LRMAC})$
   - Error to corresponding h array in $\in_{\text{LRMAC}}$
10: $\in_{\text{SPNS}}$ append ***havdist***$(T_{s,h}, \hat{\lambda}_{SPNS}, \hat{\phi}_{SPNS})$
   - Error to corresponding h array in $\in_{\text{SPNS}}$
   - ***havdist()*** *denotes the haversine distance between the observed and predicted location.*

5.6 System specifications

All the tests will be carried out on a system with the specifications listed in Table 5.

Table 5. Testing system specifications.

Description	Name	Specification
CPU	Intel i7-10700K	8 core 16 threads @ 5.1 GHz
Memory	Corsair Vengeance LPX	DDR4 @ 3600 MHz
Storage	HP EX920	NVMe M.2 SSD @ 512 GB
Software	Python	3.8.5
Database	PostgreSQL	12.6
Database plugin	PostGIS	3.1.1
Operating System	Ubuntu Desktop	20.04 LTS

6. Results

In this section, we will compare the LRMAC with the SPNS.

The LRMAC is an improved version of the LRM, where the LRMAC allows for the prediction of non-linear trajectories. In Figure 9 below, the median Haversine error is shown for both the LRM (red) and LRMAC (blue), together with the standard deviation (SD) from the median. The LRMAC has a reduced SD and median error compared to the LRM. In terms of short-term prediction accuracy, we see that the LRM and LRMAC are not significantly different, but for longer prediction windows (>120 min), the proposed method shows an improvement. To further support this observation, when looking at Figure 10, we can see that the LRMAC reproduced the trajectory, where the LRM went off course. The LRMAC assumes a constant speed calculated which was estimated from the first ω observed observations. The LRMAC predicted where the vessel would be after 8 hours, should it have traveled at a constant speed. The prediction error comparison does not accurately represent the performance of the LRMAC. When looking at the predicted trajectory, we can see that the predicted LRMAC trajectory is more similar to the actual trajectory when compared to the LRM.

The error is measured as the Haversine distance between the actual observation at time t compared to the predicted observation at time t for each method. The LRMAC has a lower SD for longer range predictions than the LRM. This is expected as our test set contains vessels that are in the highways depicted in Figure 3. Shorter prediction periods h will mean the subsets will be near-linear as cargo and tanker vessels have a slow rate of turn. We do not expect a vessel trajectory to stay linear for more extended prediction periods, as there are obstacles like landmasses and other vessels. The LRM will keep predicting linearly, whereas the LRMAC will adapt to predict non-linear trajectories using historical COG information. We can see from $h = 240$ onwards that the median difference between the LRM and LRMAC increases. We expect that the LRMAC will be able to predict with any SM that has path directionality.

In Figure 10, the LRM (lime) and LRMAC (aqua) predictions are shown of MMSI 419689000, together with the actual trajectory (magenta) that the vessel had. Looking at the error lines of the LRMAC (yellow) and the LRM (red), we see that their distances to the expected location differ by 3.47 km. The error distances are not significantly different; however, the LRMAC was still able to predict the correct path, given the constant speed assumption. The LRMAC only used $\omega = 3$ historic observations to predict the 8-hour trajectory. The background of Figure 10 depicts the vessel counts SM $\boldsymbol{K}$, shown in Figure 3. Given a use case where the historic data have paths that are longer and highly non-linear (not just a piecewise linear historic path), we believe that the LRMAC would do significantly better than the LRM as it will follow the trend of the historic data.

In Figure 11, the median prediction error of the LRMAC (blue) is compared to the SPNS (green). Similar to the LRM and LRMAC comparison, the error was calculated over all the sub trajectories with

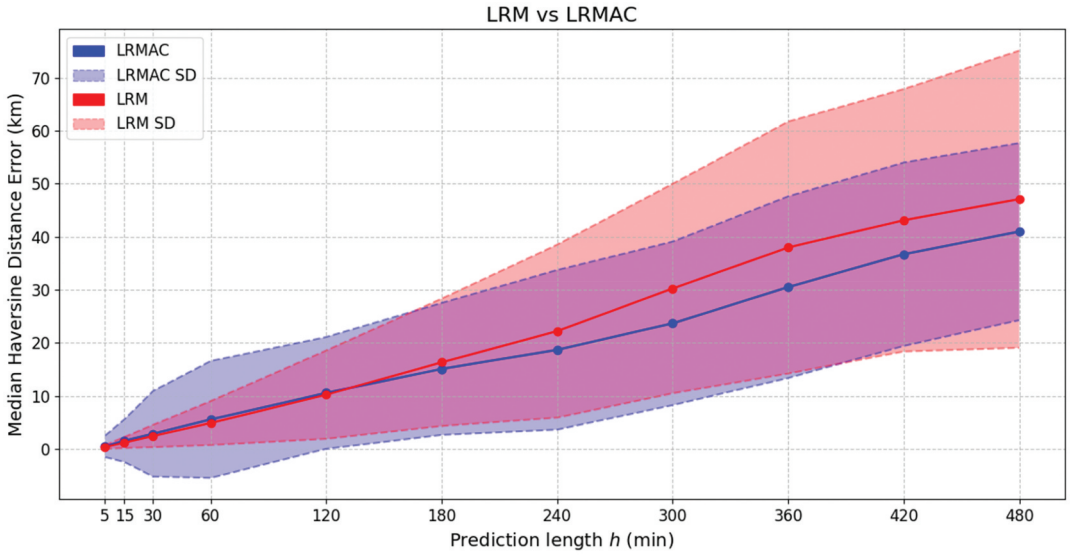

Figure 9. LRM vs LRMAC.

a prediction length of h. The median error for both methods was extracted at each h together with the standard deviation (SD). Note that the SD is plotted around the median, instead of the mean absolute deviation, and this was done to reflect the algorithmic stability.

Figure 11, which compares the LRMAC and the SPNS, is a better comparison as both the LRMAC and SPNS follow the historic route. Looking at Figure 11, we see no significant prediction performance difference between the LRMAC and the SPNS for short prediction periods. However, when considering longer prediction periods (>120 min), the median error and SD of the LRMAC are lower than that of the SPNS. The SPNS's SD starts to increase dramatically, compared to the LRMAC's SD, which increases at a lower rate. When looking at the prediction interval $[5, 15]$ min, the recommended time prediction horizon for the SPNS, the SPNS has a smaller SD than the LRMAC.

One assumption that the LRMAC makes is that a vessel keeps a constant speed, and the prediction is based on this constant speed assumption. The SOG of a vessel is derived from the last ω observed observations. The SPNS calculates the SOG based on the historic observations, but the speed is not used in the coordinate prediction step but instead in the time predictions.

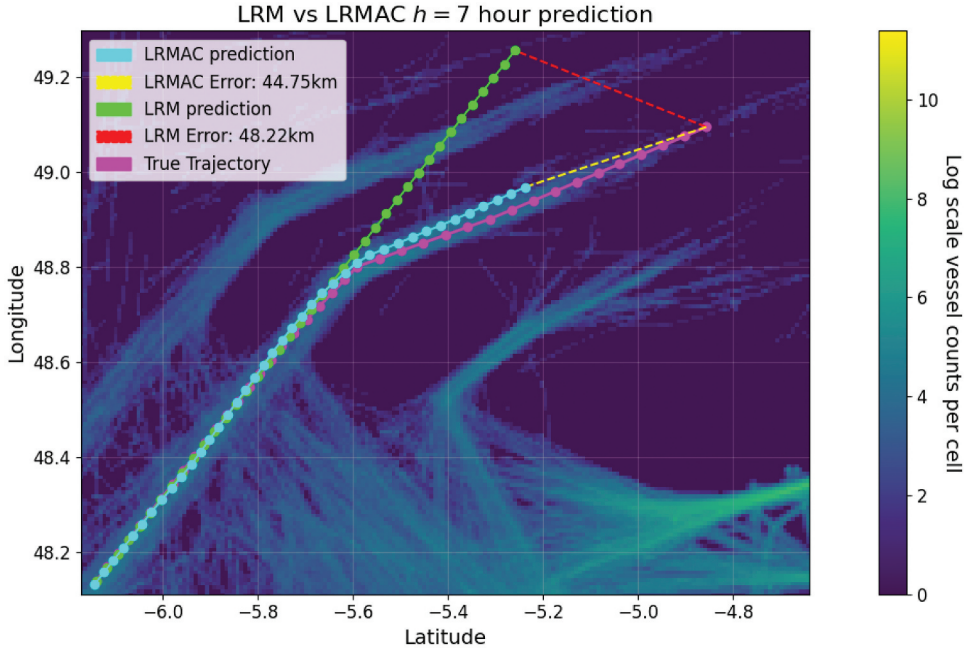

Figure 10. LRM vs LRMAC on SDM (MMSI 419689000).

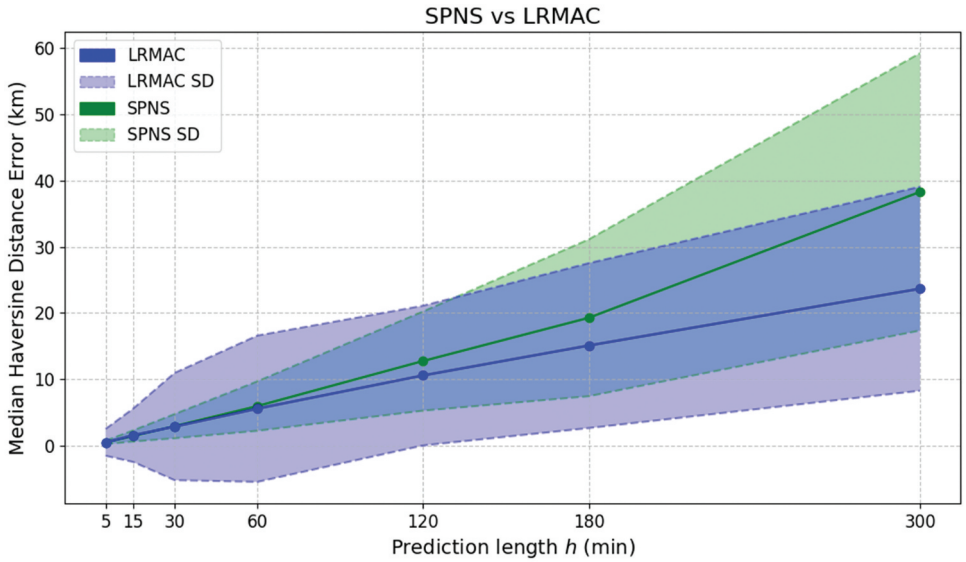

Figure 11. SPNS vs LRMAC prediction results.

In Table 6 below, we see the time complexity of both methods for each prediction length h. The time was measured over all the sub-trajectories, and the median time is shown. The SPNS computation time is significantly longer to predict than the LRMAC. The speed of the SPNS is limited by the time it takes to query the CNs from a database. Larger database table sizes will lead to more significant query times as there are more observations to search through. The LRMAC uses SMs whose sizes stay fixed, given more observations for the same longitude and latitude range. The SM sizes will only increase if the longitude and latitude range of the observed data increase.

Looking at Figure 11 and Table 6, we see that even though the two methods are not significantly different in their prediction accuracies for short prediction periods, the LRMAC had a shorter execution time. In Figure 11, we see that the SPNS has a smaller SD than the LRMAC for the $[5, 15]$ min timeframe.

We believe that in higher density areas closer to harbors, the SPNS will have more accurate prediction results as the set of CNs will only contain vessels moving in the same direction, where the LRMAC would default to the LRM as the *a-priori* COG SD $\mathbf{\Sigma}$, will be high. In areas closer to harbors, vessels tend to move slower, and prediction timeframes are usually smaller with increased AIS coverage. We think that the SPNS can be used together with the LRMAC. If the LRMAC encounters a cell with a high SD, the SPNS can be deployed until a cell with a lower SD is encountered. A hybrid approach between the SPNS and LRMAC may improve prediction times and improve prediction accuracy in areas with a significant amount of traffic in different directions.

Table 6. LRMAC vs SPNS median prediction times.

h	LRMAC time		SPNS time		Time difference
min	s	min	s	min	min
5	0.11	0.0018	30.92	0.5154	−0.5135
15	0.33	0.0055	95.68	1.2614	−1.5892
30	0.66	0.0110	193.07	3.2178	−3.2068
60	1.31	0.0218	371.87	6.1978	−6.1760
120	2.64	0.0440	712.29	11.8715	−11.8275
180	3.98	0.0663	1026.38	17.1063	−17.0399
300	6.56	0.1093	1724.46	28.7410	−28.6317

It should be noted that the LRMAC is limited by the spatial resolution of the SMs it employs. In the case of sparse historic data, the SMs generated from this data would be inaccurate, resulting in a decrease in performance of the LRMAC. Also, when the prediction time interval between two consecutive predictions is too large, the LRMAC will skip past important *a-priori* information and have inaccurate prediction results. Furthermore, the LRMAC was only tested on cargo and tanker vessels in this study, as the movement of other vessel types, such as fishing vessels are of an erratic nature and the SMs that will reflect this behavior, with high COG standard deviation.

The LRMAC can be used to predict trajectories of vessels or to impute historic trajectories. The LRMAC would be more accurate as observations are recorded, and the observations in the window size ω is updated. Currently, the predictions of the LRMAC and SPNS were done on the assumption that only the first ω observations' information will be used, simulating AIS transponders that are switched off for extended periods of time.

7. Conclusions

In this paper, we presented an extension of the LRM method proposed by Burger, Kleynhans, and Lups Grobler (2020), which uses historic AIS vessel information (particularly the historic COG information), which is robust and easy to implement.

The LRMAC implemented in this paper had a smaller incurred prediction error and associated standard deviation than the LRM implemented by Burger, Kleynhans, and Lups Grobler (2020). The LRMAC can predict a trajectory similar to the actual trajectory where the LRM goes off course. The LRMAC makes use of SMs to improve its prediction capability. The LRMAC was compared to Single-Point Neighbor Search (SPNS), which has a similar level of computational complexity and, for the use case of predicting tanker and cargo vessel trajectories up to 8 hours into the future, showed improved results both in terms of the prediction accuracy and execution time. The LRMAC can be used to predict trajectories of vessels or impute vessel trajectories. The LRMAC will have reduced prediction accuracy in high-density areas where historically vessels traveled in different directions. We think a hybrid approach between the LRMAC and the SPNS may lead to improved prediction accuracy and execution time. Future work includes exploring the possibility of a hybrid approach between the LRMAC and SPNS.

Notes

1. For more on the *International Maritime Organization (IMO),* see: www.imo.org . For additional information on AIS transponders see: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx.
2. For more on *PostgreSQL* see: www.postgresql.org
3. For more on *PostGIS* see: www.postgis.net

Acknowledgement

We would like to thank MUNUS International for all the financial support given *(munus.ai).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Christiaan Neil Burger an MSc Computer Science student at Stellenbosch University, South Africa, in 2021. The contents of this paper form part of his Masters' thesis. He is in the field of Data Science with a focus on data mining, prediction, and machine learning.

Waldo Kleynhans received a Ph.D. (Electronic Engineering) from the University of Pretoria, South Africa as well as an MBA from Heriot-Watt University, Scotland in 2012 and 2011 respectively. He is a statistical signal processing and machine learning expert focusing on various application areas including maritime domain awareness and telecommunications. He holds adjunct faculty positions with the University of Pretoria, South Africa as well as San Diego State University, USA.

Trienko Lups Grobler received his PhD in Engineering at the University of Pretoria in 2013. He was a postdoctoral fellow at Rhodes University from 2013 to 2016. He joined the Computer Science Division of Stellenbosch University in 2017. His research interests include remote sensing and interferometry.

ORCID

Christiaan Neil Burger http://orcid.org/0000-0002-5327-7324
Waldo Kleynhans http://orcid.org/0000-0002-7046-7023
Trienko Lups Grobler http://orcid.org/0000-0001-5274-0105

Data availability statement

The data that support the findings of this study are available at https://doi.org/10.5281/zenodo.1167595 (Ray et al. 2019).

References

Alizadeh, D., A. Asghar Alesheikh, and M. Sharif. 2021. "Vessel Trajectory Prediction Using Historical Automatic Identification System Data." *Journal of Navigation* 74 (1): 156–174. doi:10.1017/S0373463320000442.

Burger, C.N., W. Kleynhans, and T. Lups Grobler. 2020. "Discrete Kalman Filter and Linear Regression Comparison for Vessel Coordinate Prediction." 2020 21st IEEE International Conference on Mobile Data Management (MDM). Versailles, France: IEEE. 269–274. doi: 10.1109/MDM48529.2020.00062.

Chen, X., Q. Lei, Y. Yang, Q. Luo, O. Postolache, J. Tang, and W. Huafeng. 2020. "Video-Based Detection Infrastructure Enhancement for Automated Ship Recognition and Behavior Analysis." *Journal of Advanced Transportation* 2020: 1–12. doi:10.1155/2020/7194342.

Chen, X., S. Wang, C. Shi, W. Huafeng, J. Zhao, and F. Junjie. 2019. "Robust Ship Tracking via Multi-view Learning and Sparse Representation." *Journal of Navigation* 72 (1): 176–192. doi:10.1017/S0373463318000504.

Dimitrios, Z., E.K. Xidias, and D. Lekkas. 2016. "Real-time Vessel Behavior Prediction." *Evolving Systems* 7 (1): 29–40. doi:10.1007/s12530-015-9133-5.

Ester, M., H.-P. Kriegel, J. Sander, and X. Xiaowei. 1996. "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 226–231.

Forti, N., L. M. Millefiori, P. Braca, and P. Willett. 2020. "Prediction of Vessel Trajectories from AIS Data via Sequence-to-Sequence Recurrent Neural Networks." In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8936–8940. doi: 10.1109/ICASSP40776.2020.9054421.

Grobler, T.L., and W. Kleynhans. 2019. "Extracting High-Volume Traffic Routes from AIS Spatial Distribution Maps." In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, 10031–10034. doi: 10.1109/IGARSS.2019.8900105.

Grünwald, P.D., I. Jae Myung, and M.A. Pitt. 2005. *Advances in Minimum Description Length: Theory and Applications*. Cambridge, MA: MIT Press.

Hexeberg, S., A. L. Flåten, and E. F. Brekke. 2017. "AIS-based Vessel Trajectory Prediction." In *20th International Conference on Information Fusion (Fusion)*, 1–8. doi: 10.23919/ICIF.2017.8009762.

Jaskolski, K. 2017. "Automatic Identification System (AIS) Dynamic Data Estimation Based on Discrete Kalman Filter (KF) Algorithm." *Maritime Technical Journal* 211 (4): 71–87. doi: 10.5604/01.3001.0010.6747.

Jiashun, C. 2012. "A New Trajectory Clustering Algorithm Based on TRACLUS." In *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*, 783–787. doi: 10.1109/ICCSNT.2012.6526048.

Kalman, R.E. 1960. "A New Approach to Linear Filtering and Prediction Problems." *Journal of Basic Engineering* 82 (1): 35–45. doi:10.1115/1.3662552.

Lee, J.-G., J. Han, and K.-Y. Whang. 2007. "Trajectory Clustering: A Partition-and-GroupFramework." Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. Beijing, China: Association for Computing Machinery. 593–604. doi: 10.1145/1247480.1247546.

Murray, B., and L. Prasad Perera. 2020. "A Dual Linear Autoencoder Approach for Vessel Trajectory Prediction Using Historical AIS Data." *Ocean Engineering* 209: 107478. doi:10.1016/j.oceaneng.2020.107478.

Pallotta, G., S. Horn, P. Braca, and K. Bryan. 2014. "Context-enhanced Vessel Prediction Based on Ornstein-Uhlenbeck Processes Using Historical AIS Traffic Patterns: Real-World Experimental Results." In *17th International Conference on Information Fusion (FUSION)*,1–7. IEEE.

Pallotta, G., M. Vespe, and K. Bryan. 2013a. "Traffic Route Extraction and Anomaly Detection from AIS Data." International COST MOVE Workshop on Moving Objects at Sea. Brest, France.

Pallotta, G., M. Vespe, and K. Bryan. 2013b. "Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction." *Entropy* 15 (6): 2218–2248. doi:10.3390/e15062218.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, G. Olivier, M. Blondel, et al. 2011. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12:2825–2830.

Ray, C., R. Dréo, E. Camossi, A.-L. Jousselme, and C. Iphar. 2019. "Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance." *Data in Brief* 25: 104141. doi:10.1016/j.dib.2019.104141. Accessed 2020.

Rong, H., A.P. Teixeira, and C. Guedes Soares. 2019. "Ship Trajectory Uncertainty Prediction Based on a Gaussian Process Model." *Ocean Engineering* 182: 499–511. doi:10.1016/j.oceaneng.2019.04.024.

Tang, H., Y. Yin, and H. Shen. 2019. "A Model for Vessel Trajectory Prediction Based on Long Short term Memory Neural Network." *Journal of Marine Engineering & Technology* 21 (3): 136–145. doi:10.1080/20464177.2019.1665258.

Wijaya, W.M., and Y. Nakamura. 2013. "Predicting Ship Behavior Navigating through Heavily Trafficked Fairways by Analyzing AIS Data on Apache HBase." First International Symposium on Computing and Networking. Matsuyama: IEEE. 220–226. doi: 10.1109/CANDAR.2013.39.

Xiao, Z., F. Xiuju, L. Zhang, and R. Siow Mong Goh. 2020. "Traffic Pattern Mining and Forecasting Technologies in Maritime Traffic Service Networks: A Comprehensive Survey." *IEEE Transactions on Intelligent Transportation Systems* 21 (5): 1796–1825. doi:10.1109/TITS.2019.2908191.

Xin, X. 2020. "Context-Based Trajectory Prediction with LSTM Networks." CIIS 2020: 2020 The 3rd International Conference on Computational Intelligence and Intelligent Systems. New York, NY, USA: Association for Computing Machinery. 100–104. doi: 10.1145/3440840.3440842

Yaun, Z., J. Liu, Y. Liu, and L. Zongzhi 2019. "A Novel Approach for Vessel Trajectory Reconstruction Using AIS Data." In *29th International Ocean and Polar Engineering Conference*, 4554–4559.