Ocean Engineering 209 (2020) 107478

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data

Brian Murray [*], Lokukaluge Prasad Perera

UiT The Arctic University of Norway, Tromsø, Norway

ARTICLE INFO

Keywords:
Maritime situation awareness
Trajectory prediction
Collision avoidance
Machine learning
Autoencoder
AIS

ABSTRACT

Advances in artificial intelligence are driving the development of intelligent transportation systems, with the purpose of enhancing the safety and efficiency of such systems. One of the most important aspects of maritime safety is effective collision avoidance. In this study, a novel dual linear autoencoder approach is suggested to predict the future trajectory of a selected vessel. Such predictions can serve as a decision support tool to evaluate the future risk of ship collisions. Inspired by generative models, the method suggests to predict the future trajectory of a vessel based on historical AIS data. Using unsupervised learning to facilitate trajectory clustering and classification, the method utilizes a cluster of historical AIS trajectories to predict the trajectory of a selected vessel. Similar methods predict future states iteratively, where states are dependent upon the prior predictions. The method in this study, however, suggests predicting an entire trajectory, where all states are predicted jointly. Further, the method estimates a latent distribution of the possible future trajectories of the selected vessel. By sampling from this distribution, multiple trajectories are predicted. The uncertainties of the predicted vessel positions are also quantified in this study.

1. Introduction

As more advanced technologies are introduced into transportation systems, the opportunity to enhance the safety of these systems increases. Increased computational power in conjunction with advances in artificial intelligence, and the ubiquity of sensor data, allow for new methods to be implemented across a wide number of sectors. Some argue that an industrial revolution is taking place, naming it Industry 4.0 (Hermann et al., 2016). The automotive industry is an example of a sector in which such technological advances are embraced and integrated into existing systems. The shipping industry, however, has historically been more conservative in adopting new technologies, often relying on older, but proven systems. Nonetheless, advances are being made, with some arguing that shipping is also undergoing a technological revolution, Shipping 4.0 (Rødseth et al., 2015).

1.1. Maritime situation awareness

An essential aspect of Shipping 4.0 is arguably implementing modern technologies to enhance the safety of maritime operations. Effective collision avoidance strategies are an integral part of maintaining safe operations. The efficacy of such strategies relies on the degree of situation awareness of the navigator. Situation awareness was defined in Endsley et al. (2003) as *"being aware of what is happening around you and understanding what that information means to you now and in the future"*. Endsley (1995) separated situation awareness into three levels:

1. Perception of the elements in the current situation
2. Comprehension of the current situation
3. Projection of the future status

Maritime situation awareness largely relates to obstacle detection and prediction of close-range encounter situations. Such obstacles will primarily be other vessels, referred to as target vessels in an encounter situation. It is, therefore, essential for a navigator to have an adequate degree of situation awareness in order to conduct effective collision avoidance maneuvers. A thorough review of collision avoidance methods can be found in Tam et al. (2009).

The main challenge for a navigator is determining the risk of collision based on their degree of situation awareness. In light of Endsley's definition, one can classify level one situation awareness as relating to obstacle detection, and level two an evaluation of the current collision risk. Perera and Guedes Soares (2015) addressed collision risk detection and quantification techniques with respect to integrating modern technologies. The study discussed the concept of *e-Navigation* as introduced by the International Maritime Organization. *e-Navigation* aims to utilize maritime information by electronic means to enhance

[*] Corresponding author.
E-mail address: brian.murray@uit.no (B. Murray).

https://doi.org/10.1016/j.oceaneng.2020.107478
Received 12 November 2019; Received in revised form 31 March 2020; Accepted 30 April 2020

Nomenclature

$\mathbf{a}$	Arbitrary AIS Parameter Vector
$\mathbf{A}$	Set of AIS Data
$\mathbf{b}$	Backward Trajectory Feature Vector
$\mathbf{C}$	Data Cluster
d	Euclidean Distance
$\mathbf{e}$	Eigenvector
$\mathbf{E}$	Eigenvector Matrix
$\mathbf{f}$	Forward Trajectory Feature Vector
$\mathbf{h}$	Latent Feature Vector
$\mathbf{H}$	Distribution of Latent Features
k	Number of Nearest Neighbors
l	Number of Eigenvectors
L	Number of Data Points in Selected Trajectory
N	Number of Trajectories
M	Number of Models in Mixture Model
$\mathbf{p}$	Selected Vessel Position Vector
$\mathbf{P}$	Distribution of Selected Vessel Position
$\mathbf{s}$	Vessel State Vector
$\mathbf{S}$	Scatter Matrix
T	Time Horizon [min]
v	Speed over Ground [m/s]
w	Interpolation Weight
x	UTM x-coordinate [m]
y	UTM y-coordinate [m]
$\boldsymbol{\Lambda}$	Eigenvalue Matrix
$\boldsymbol{\mu}$	Mean Vector
π	Prior Distribution
σ	Standard Deviation
$\boldsymbol{\Sigma}$	Covariance Matrix
χ	Course over Ground [°]

Subscripts

0	Initial State
b	Backward Trajectory
bc	Between-class
wc	Within-class
f	Forward Trajectory
i	Sample Number
j	jth State
m	Model Number in GMM
p	Trajectory Prediction
s	Selected Vessel
δ	Maximum Offset

Superscripts

$\hat{}$	Estimated Parameter/State

the safety of maritime operations. It was argued that integrated bridge systems are the main focus. Here, relevant information relating to ship navigation should be properly integrated in order to provide decision support to navigators. As such, Perera and Guedes Soares (2015) argued that the best navigation tools possible should be available on board the vessel to aid the navigator in identifying high risk situations. Based on this risk evaluation, adequate collision avoidance maneuvers can be conducted that adhere to the COLREGS as outlined in Perera et al. (2010).

A wide range of technologies are currently adopted to aid in providing situation awareness to navigators, including radar, conning and ECDIS (Electronic Chart Display and Information System). Radar systems facilitated by ARPA (Automatic Radar Plotting Aid) and the ECDIS are essential in aiding navigators to determine the risk of collision. Generally, the future state of a target vessel is estimated based on calculations of constant course and speed values. These estimates can then be used by the navigator to estimate collision risk parameters relating to the closest point of approach (CPA), such as the time (TCPA) and distance (DCPA). Based on this information, a navigator can make a decision with respect to a potential collision situation. However, predicting collision situations far in advance, i.e. level three of Endsley's situation awareness model, will be the focus area of this study.

1.2. Vessel trajectory prediction

Predicting ship behavior as in Perera (2017) can provide decision support to navigators to make appropriate collision avoidance maneuvers. Advanced techniques, e.g. Perera et al. (2012), where extended Kalman filters were utilized to estimate ship trajectories, can further enhance the situation awareness of navigators. Such methods, however, are only useful for prediction horizons in the order of seconds to minutes. As such, they will only aid navigators in cases in which close-range encounter situations are imminent. As a result, it was suggested in Perera and Murray (2019) to introduce an advanced ship predictor. This study focused on methods to provide autonomous vessels with adequate situation awareness. However, such methods are also relevant for use in decision support to ship navigators. In this approach, a local and global scale ship predictor were suggested. At a local scale, techniques such as those outlined in Perera (2017) can be utilized to aid in short term trajectory predictions in order to aid in effective collision avoidance maneuvers once a collision is deemed imminent. On the global scale however, long term trajectory predictions, on the scale of 5–30 min, are conducted. Such predictions aim to prevent close-range encounter situations from occurring at all. Such predictions are, however, not straight forward, as the future intentions of the vessel are unknown, and may potentially be complex.

1.2.1. AIS based vessel trajectory prediction

One method to conduct vessel trajectory predictions on a global scale is to utilize historical AIS (Automatic Identification System) data. By exploiting AIS data, insight into historical ship behavior can be gained. Multiple ship parameters relating to historical ship movement are stored in databases, available for use. Such parameters include the position, speed and course over ground. Recently, there has been a significant increase in research into exploiting AIS data for maritime situation awareness. A number of studies have focused on evaluating grouping trajectories together to gain insight into maritime patterns. Aarsæther and Moan (2009) for instance utilized computer vision techniques to group trajectories and subsequently calculate statistics for each traffic pattern. Zhang et al. (2018) also utilized AIS data via a data driven approach that compressed and clustered trajectories to extrapolate the general behavior patterns of vessels traveling along the same route. Subsequently, given a starting point, the Ant Colony Algorithm was utilized to output an optimal route to the destination. Zhang and Meng (2019) also presented a data driven method to determine a probabilistic ship domain based on AIS data. Such ship domains can subsequently be utilized for collision risk assessment. A comprehensive review of various methods to exploit AIS data for maritime navigation was presented in Tu et al. (2017).

Of primary interest for this study, however, is the work done to predict the future trajectory of a vessel that can be utilized in a global scale ship predictor. As such, the aforementioned methods are of limited usefulness. Ristic et al. (2008) utilized a particle filter to predict the future behavior of vessels using historical AIS data, but the predicted future positions had a large uncertainty associated with them, making the method of limited use with respect to collision avoidance decisions and actions. A number of studies also have focused on clustering historical trajectories, and subsequently classifying a vessel to one of these groups. Pallotta et al. (2013) for instance presented the TREAD (Traffic Route Extraction and Anomaly Detection) methodology that clustered all historical trajectories in a specific region to identify traffic routes and subsequently classify a partial trajectory to one of these routes for anomaly detection. The method also addressed assessing the probability of a position along a route. Pallotta et al. (2014) further expanded upon the TREAD methodology by predicting the vessel position along a route using the Ornstein–Uhlenbeck stochastic process. The TREAD technique, however, clustered entry points, waypoints and stationary points of trajectories within a defined region. In this respect, the trajectory through the entire region was utilized to group similar trajectories together. This can result in trajectories with large differences between sub-trajectories being clustered together. For predictions in the order of hours, this is not an issue, and the outlined method is quite effective. For collision avoidance purposes, a higher fidelity prediction is required that requires more discrimination between trajectories. Other studies on clustering and classification include Zhao and Shi (2019) which clustered trajectories by using dynamic time warping and the Douglas Peucker algorithm, in addition to Zhou et al. (2019) which clustered using k-means, and subsequently classified ship behavior. Methods relying on dynamic time warping and way-point based clustering will cluster trajectories based on similar spatial behavior, but be invariant with respect to time. As such, trajectories that have similar spatial shapes will be grouped together despite various behavior being observed at different relative times. This may be detrimental to a subsequent trajectory prediction, in that the clustering capability is restricted to the shapes of trajectories, irrespective of their duration and potential differences in sub-trajectories.

Mazzarella et al. (2015) also presented a trajectory prediction approach using AIS data, via a Bayesian network approach with a particle filter. This method was designed for predictions in the order of hours, and as such of limited use with respect to collision avoidance. Other methods include Hexeberg et al. (2017), where a Single Point Neighbor Search method was presented based on historical AIS data. The method does not involve any clustering or classification steps, and as such suffers when handling branching. Dalsnes et al. (2018) built upon this work and provided multiple predictions using a prediction tree. This approach allows for a probability estimate of a future prediction to be estimated using a Gaussian mixture model. These methods, however, do not utilize the relationship between data points, as future states are based solely on the neighborhood of previous states which may not have any relationship to the prior predicted states. This will have a negative effect on the accuracy. Rong et al. (2019) also presented a probabilistic trajectory prediction method using a Gaussian Process model. This method, in addition to predicting the future position of a vessel, gave an uncertainty estimate associated with the prediction. The method had good results for the regular trajectories investigated off the coast of Portugal, but did not address how to deal with more complex traffic situations and trajectories, which likely will degrade the outcome.

1.3. Generative models

The method utilized in this study takes an alternative approach to those that come before. It is inspired by a field of deep learning known as generative models (Foster, 2019), widely adopted in the field of machine learning. Such models have recently gained a high degree of popularity due to the powerful generative ability of deep learning models. One such general model is the autoencoder. An autoencoder is a type of neural network, with its most simple form being a multi-layer perceptron (Bourlard and Kamp, 1988). The objective of an autoencoder is to reconstruct the data fed into it, essentially copying its input to its output. Such techniques are, however, not extensively applied in the maritime domain. Some studies have looked into applying these approaches in the maritime domain, e.g. Perera and Mo (2018), where autoencoders were suggested as a tool to compress data to facilitate more effective maritime data transmission. Autoencoders are considered to have two parts: an encoder function $f(\mathbf{x})$ that produces the code, $\mathbf{h}$, shown in (1), and a decoder function, $g(\mathbf{h})$, that reconstructs the data from the code shown in (2). An integral part of an autoencoder is the internal hidden layer, $\mathbf{h}$, that represents the code space, often referred to as the latent representation of the data.

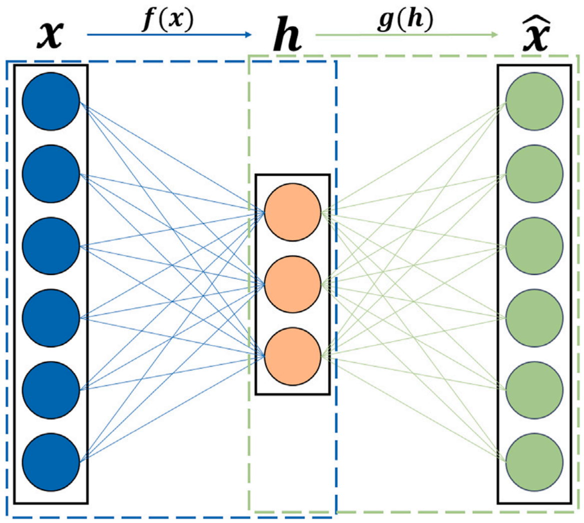

Fig. 1. Autoencoder architecture with the encoder illustrated by the blue dashed box, and the decoder by the green dashed box-. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\mathbf{h} = f(\mathbf{x}) \tag{1}$$

$$\hat{\mathbf{x}} = g(\mathbf{h}) \tag{2}$$

For an autoencoder to be useful, it must provide a form of functionality other than mapping the input to the output. Undercomplete autoencoders (Goodfellow et al., 2016), i.e. where $\mathbf{h}$ has a smaller dimension than $\mathbf{x}$, provide a bottleneck in the code space through which the network can learn a meaningful latent representation of the data. The mapping function of the input data to the code space, $f(\mathbf{x})$, can be thought of as a data compression operation, or parameter reduction. The encoder strives to create a meaningful latent representation that preserves as much information as possible, such that the decoder has adequate information to reconstruct the data. As such, when an autoencoder is trained on a dataset, it will adapt such that the encoder preserves the most important information in the dataset.

Traditionally, autoencoders have been utilized for dimensionality reduction and feature generation (Goodfellow et al., 2016). In this case, the latent representation can be utilized for data visualization or to generate more relevant features for further processing. Additionally, once an autoencoder is trained, data can be compressed and stored. Subsequently, it can be decoded for later use. Such applications are often very useful. However, the generative capabilities of autoencoders have recently also gained interest. Alternatively to encoding and decoding the data, one can solely utilize the decoder in order to generate new data. This is done by sampling a data point in the code space, and subsequently running a forward pass through the decoder to reconstruct the data. In this manner, one can interpolate between existing data points in the code space to generate new samples.

The variational autoencoder (Kingma and Welling, 2014; Rezende et al., 2014) is a popular type of generative model. A variational autoencoder is a probabilistic version of an autoencoder where the network learns a probability distribution of the reconstructed data based on a learned distribution over the code. In this manner, there is a continuous distribution in the code space that can be sampled from. Kingma and Welling (2014) investigated the use of a variational autoencoder and presented a figure illustrating generated images from a variational autoencoder trained on the MNIST dataset of handwritten digits. The figure illustrates the interpolation of the digits with a 2-D latent code. Each axis along the figure can be thought of as one dimension in the latent code. It is evident that as one moves around within the code space, the digits morph from one digit to another. The latent representation is able to capture the most important differences in the data along the respective axes. As such, one can generate a new image simply by interpolating within the code space generated by training the autoencoder.

Such generative capabilities can be extended to virtually any dataset, where an autoencoder is trained, and based on the latent distribution of the data, can generate new data samples from the distribution. As such, if an autoencoder is trained on a cluster of trajectories, it should be able to generate a new trajectory by interpolating in the latent space.

1.4. Contribution

The objective of this study is to provide an architecture that can support collision avoidance actions by providing situational awareness to navigators or autonomous agents. As a result, the architecture differs from that of similar studies with respect to its design. To aid in situation awareness, a method is suggested to provide a global scale ship predictor that estimates the future 30 min trajectory of a selected vessel with a high degree of fidelity. As opposed to a number of other studies, the approach in this study is designed to run live, i.e. without any pre-trained models. A ship in any region, given an adequate density of historical AIS data can, therefore, utilize the developed architecture. In the suggested approach, relevant historical ship trajectories are extracted from an AIS database, that represent the possible future 30 min behavior of a selected vessel. This dataset comprises only relevant data with respect to the observed state of a selected vessel for the purpose of trajectory prediction, and as such provides the basis for the remainder of the prediction methodology. Inherent differences in behavior are described by these trajectories, which in turn represent the possible modes of the future 30 min behavior the selected vessel may belong to. Therefore, the trajectory representation differs from other methods that evaluate entire trajectories for a region. The representation in this study provides higher fidelity predictions as a result.

In order to discover clusters of similar trajectories, other approaches utilize trajectory representations that introduce invariance with respect to time, e.g. dynamic time warping, or point based techniques using waypoints. These techniques are effective for clustering trajectories of similar shapes together. For the purpose of this study however, it is of interest to discover all possible trajectory modes that represent the future 30 min behavior of the selected vessel, not just trajectories of similar shapes for the region. As such, trajectories should not be invariant with respect to time. Therefore, by representing each trajectory by vectors of equal length containing the future 30 min of trajectory data, the representations will be sensitive to the time at which various behavior is observed. Such a representation will, therefore, be more sensitive to modes within primary ship routes. Discovering these modes will provide a much better basis for a subsequent trajectory prediction for collision avoidance purposes, as the prediction must be as accurate as possible. This study suggests to cluster compressed trajectories via Gaussian Mixture Models to an unspecified number of clusters, each representing a mode of future behavior, and is shown to have good performance for the purpose of the study.

Once a selected vessel is classified to a given cluster of historical AIS behavior, this data is used directly in the dual linear autoencoder prediction architecture. This architecture differs significantly from other methods, which generally predict future states in manner such that they are predicated upon previous predicted states. In this study, it is suggested to predict entire trajectories, i.e. all future states are predicted jointly. A novel trajectory prediction technique inspired by generative models is, therefore, suggested using a dual linear autoencoder approach. In this approach, a latent representation of the possible future behavior of the selected vessel is calculated. The latent representation can be viewed as an encoded version of the data. Using this distribution, the encoded representation of the selected vessel's future behavior is estimated by interpolating between the encoded data points. By decoding the estimate of the latent representation of the future trajectory, an entire trajectory is predicted by a single matrix multiplication operation. Other methods predict an average of the behavior in the cluster, i.e. the average of the distribution, whereas the method suggested in this study will estimate the most likely sample. As such, the prediction is discrete, and can provide more accurate predictions than other methods in which the behavior is averaged out.

The prediction accuracy will also be enhanced for clustering schemes that are able to identify ship modes with a high degree of fidelity, as clusters that contain multiple ship modes will result in the prediction averaging out the behavior between modes due to the interpolation. As a result, the overall architecture of the study allows for higher fidelity predictions than other methods. Additionally, the study provides a method to estimate the distribution of the selected vessel's future trajectory latent representation. This is to account for uncertainty in the estimate, and by decoding samples from this distribution, a region of uncertainty for the predicted position at a given prediction horizon can be evaluated. The suggested architecture also utilizes linear autoencoders. Therefore, it allows for fast predictions as they are facilitated by calculating eigenvectors, and conducting subsequent matrix multiplications. As such, there is no training of a deep neural network. This architecture is, therefore, ideal for live predictions, as the calculations involved in the prediction itself will be fast. This approach in this study, therefore, provides a method to conduct live predictions of higher fidelity with respect to collision avoidance purposes on a global scale than other methods, as well as an effective method to quantify the uncertainty of the predicted positions.

2. Methodology

In this section, the methodology utilized to predict the future trajectory of a selected vessel is outlined. The objective of the method is to accurately predict the future trajectory of a selected vessel, and provide an uncertainty estimate with respect to the predicted positions. The overall architecture of the method is illustrated in Fig. 2. The method can be separated into three modules. The first is the trajectory clustering module, where groupings of similar historical trajectories are discovered. It is assumed that the future trajectory of a selected vessel can be inferred based on the historical trajectories of other vessels in the region. As such, the selected vessel is classified to one of the discovered clusters in the trajectory classification module. Based on the cluster of trajectories to which the selected vessel is classified, a trajectory prediction is conducted in the trajectory prediction module. This is achieved via a novel dual linear autoencoder approach. In this approach, two linear autoencoders are utilized. The forward linear autoencoder provides a latent representation of the historical trajectories that can be used to infer the future trajectory of the selected vessel. The backward linear autoencoder provides a latent representation of the prior behavior of the historical trajectories. Based on a similarity measure evaluated in the latent space of the backward linear autoencoder, a latent interpolation is conducted to estimate the forward latent representation of the selected vessel. Subsequently, this estimate can be decoded, resulting in a trajectory prediction.

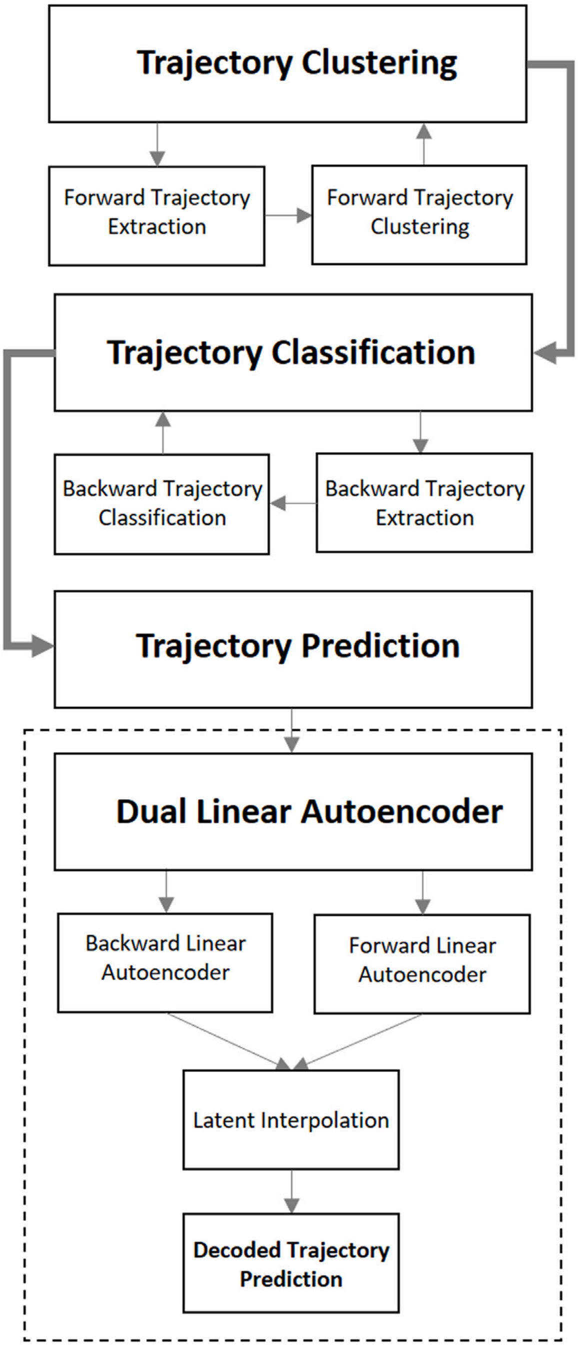
Trajectory Clustering
Forward Trajectory Extraction
Forward Trajectory Clustering
Trajectory Classification
Backward Trajectory Classification
Backward Trajectory Extraction
Trajectory Prediction
Dual Linear Autoencoder
Backward Linear Autoencoder
Forward Linear Autoencoder
Latent Interpolation
Decoded Trajectory Prediction

Fig. 2. Prediction architecture, with the dual linear autoencoder block within the dashed lines.

2.1. Unsupervised trajectory clustering and classification

In this section, the methodology involved in clustering historical AIS trajectories and classifying the trajectory of a selected vessel is outlined. This work in this section builds upon preliminary work described in Murray and Perera (2019). The reader is, therefore, referred to Murray and Perera (2019) for further details. It can be argued that investigating the historical behavior of vessels in a particular geographical region can provide insight into the future behavior of a vessel observed in that region. However, historical vessel trajectories will have a high degree of variation. This variation is due to the existence of multiple traffic routes, as well as the characteristics of the vessel with respect to the speed it will traverse along a given route. It is, therefore, of interest to identify groupings of similar trajectories, such that specific traffic behavior can be identified. Once such groupings are identified, a selected vessel can be classified as belonging to a given group. In this manner, a subsequent trajectory prediction can be conducted on an enhanced data set, where the data used for prediction will likely have a high degree of similarity to that of the selected vessel. This can be thought of as advanced form of preprocessing of the AIS data, such that subsequent trajectory predictions will have a higher degree of accuracy.

Grouping such data can be conducted via a technique from the field of machine learning known as clustering. This is a form of unsupervised learning, where labels for the data are unavailable. Clustering has as its goal to discover underlying groupings in the data, i.e. identify clusters of data. Once the historical vessel trajectories have been clustered, the observed trajectory can be used to classify the selected vessel to one of the discovered clusters.

2.1.1. Trajectory extraction

The initial state of a selected vessel is defined in (3). This state represents the observed parameters of the selected vessel available via the on-board sensor suite of the own-ship. The parameters in this state provide the basis for the selection of relevant historical ship trajectories for a subsequent prediction of the selected vessel's future trajectory.

$$\mathbf{s}_0 \rightarrow [x_0, y_0, \chi_0, v_0] \tag{3}$$

The method first identifies historical AIS data points with a high degree of similarity to $\mathbf{s}_0$. In essence, this means that it is desirable to identify ships that were at a similar position, with a similar course and speed, at some point in history. In order to achieve this, an initial cluster $\mathbf{C}_0$ is created. $\mathbf{C}_0$ is defined to be a rectangular cluster orthogonal to χ_0, with a height of δ_H parallel to χ_0, and a width δ_W orthogonal to χ_0. z' is the rotated space with the orthogonal vectors in the original space as basis vectors. $\mathbf{C}_0$ is defined according to the following equation (Murray and Perera, 2019):

$$\begin{aligned}\mathbf{C}_0 = \{&\mathbf{a}_i \in \mathbf{A} : (|x_{z'i} - x_{z'0}| \leq \delta_W \ \wedge \ |y_{z'i} - y_{z'0}| \leq \delta_H) \\ &\wedge (|\chi_i - \chi_0| \leq \chi_\delta \ \wedge \ |v_i - v_0| \leq v_\delta)\}\end{aligned} \tag{4}$$

Additionally, data points that do not match the ship type of the selected vessel are removed. $\mathbf{C}_0$ will, however, likely contain multiple data points from the same trajectory. As such, unique trajectories are identified, and the most similar point to $\mathbf{s}_0$ in each unique trajectory determined. $\mathbf{C}_0$ is then updated by filtering out all data points other than these most similar points. In this manner, $\mathbf{C}_0$ only contains one data point per trajectory.

Once the initial clustering phase is completed, a forward and backward trajectory extraction operation is conducted. This entails that for all trajectories in $\mathbf{C}_0$, the forward trajectory from the corresponding point in $\mathbf{C}_0$ is extracted. This can be thought of as the future trajectory defined in relation to the point in $\mathbf{C}_0$. The length of the extracted forward trajectory is defined based on the desired prediction horizon, T_p. For instance, if a 30 min prediction is desired, 30 min of the forward trajectory will be extracted. Similarly, the backward, i.e. past, trajectory from its corresponding point in $\mathbf{C}_0$ of a length corresponding to T_b into the past is extracted. Both the forward and backward trajectories are subsequently interpolated at 30 s intervals for comparative analysis. As such, each trajectory will have $L = 2 \times T_p$ entries, where each entry can be used to compare positions at a given time instance defined from the origin of the trajectory (see Fig. 3).

2.1.2. Trajectory clustering

One of the objectives of extracting the forward trajectories is to provide a dataset upon which one can identify possible future routes that the selected vessel may follow. It is, therefore, desirable to group, or cluster, these trajectories such that each possible route can be evaluated individually, as there may be many possible future routes that the selected vessel may follow. This is conducted by first generating

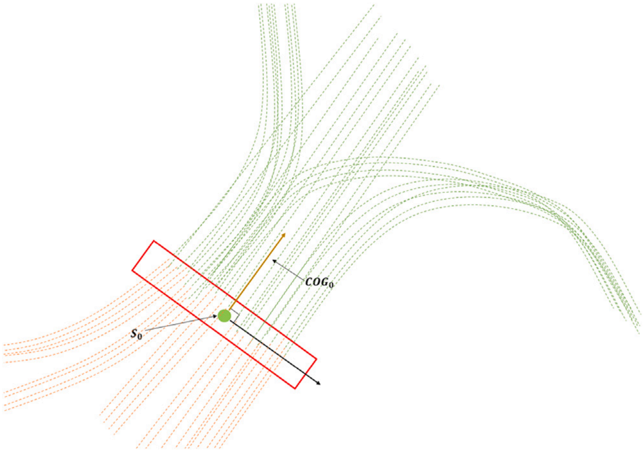

Fig. 3. $\mathbf{C}_0$ illustrated as a rectangular red box. The forward and backward extracted trajectories illustrated as green and orange lines respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

features for each trajectory by flattening the positional data given in UTM-coordinates, i.e. $[x, y]$, into a feature vector.

Clustering is a technique that groups data points based on some similarity measure, i.e. data points that are closer to one another in some n-dimensional space are more likely to be considered part of the same cluster. If the dimensionality of the space is large however, the clustering algorithm may suffer due to the curse of dimensionality discussed in Steinbach et al. (2004). One aspect of curse of dimensionality relates to points getting lost in the space due to large distances between points with respect to certain dimensions. This can make clustering in a high dimensional space challenging. Dimensionality reduction is, therefore, conducted for each trajectory via the Karhunen–Loéve (KL) transform (Karhunen, 1946) in (5), where the dimensionality is reduced from $2L$ to l.

$$\mathbf{h_f} = \mathbf{E}^T \mathbf{f} \tag{5}$$

where $\mathbf{h_f} \in \mathbb{R}^{l\times 1}$, $\mathbf{f} \in \mathbb{R}^{2L\times 1}$ and $\mathbf{E} \in \mathbb{R}^{2L\times l}$

$$\boldsymbol{\Sigma} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T \tag{6}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{2L\times 2L}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{2L\times 2L}$

The next step is to cluster the forward trajectories. This is conducted using Gaussian Mixture Model (GMM) clustering via the Expectation Maximization (EM) algorithm. A Gaussian Mixture Model (Reynolds et al., 2000) assumes that data is comprised of a mixture of M different Gaussian distributions, each with their own mean vector $\boldsymbol{\mu}_m$, covariance matrix $\boldsymbol{\Sigma}_m$ and prior distribution $\boldsymbol{\pi}_m$. Each data point representing a forward trajectory will be clustered to the distribution of the highest probability. The EM algorithm updates the underlying parameters until a model of best fit is discovered. The assumed number of underlying distributions, M, is also varied to discover the most likely mixture. For more details on GMM clustering of trajectories, see Murray and Perera (2019).

2.1.3. Trajectory classification

Once the forward trajectories have been clustered, it is desirable to classify the selected vessel to one of the discovered clusters. One method to achieve this is to investigate the backward trajectories. Assuming that the past behavior of the selected vessel is available for a period corresponding to T_b, one can compare the past behavior of the selected vessel to the backward trajectories extracted from $\mathbf{C}_0$. The aforementioned backward and forward trajectories are in fact one single trajectory, but the forward trajectories are the section corresponding to the future behavior, and the backward trajectory the past behavior. In the classification module, the extracted backward trajectories are assigned the class labels of their corresponding forward trajectories, discovered in the clustering module. These are utilized to classify the observed trajectory of the selected vessel to one of these classes. In order for the classification process to be as effective as possible, it is of interest to generate optimal features to represent the backward trajectories. This is achieved via Linear Discriminant Analysis (LDA) (Fisher, 1936). LDA requires that the data points are labeled, and as such, the backward trajectories are given the labels of the corresponding forward trajectories. The transformation is conducted via (7). Subsequently, a classifier of choice can be utilized to classify the transformed backward trajectory of the selected vessel to one of the clusters. This will yield the most likely future route that the selected vessel will follow. In this study, a kNN classifier is utilized.

$$\mathbf{h_b} = \mathbf{E}^T \mathbf{b} \tag{7}$$

where $\mathbf{h_b} \in \mathbb{R}^{l\times 1}$, $\mathbf{b} \in \mathbb{R}^{2L\times 1}$ and $\mathbf{E} \in \mathbb{R}^{2L\times l}$

$$\mathbf{S_{wc}^{-1}S_{bc}} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T \tag{8}$$

where $\mathbf{S_{wc}^{-1}S_{bc}} \in \mathbb{R}^{2L\times 2L}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{l\times l}$.

2.2. Dual linear autoencoder

This study introduces a novel dual linear autoencoder trajectory prediction method that is further described in this section. The motivation is to predict the future trajectory of a selected vessel. The method is inspired by the generative models addressed in Section 1.3. If one can create a latent distribution of possible future trajectories, one can then interpolate between existing trajectories in the latent space, and generate a new trajectory that corresponds to the selected vessel.

Autoencoders generally have non-linear activation functions. However, the linear autoencoders investigated in this study do not have non-linear activation functions in the network, and as such the encoder and decoder functions will simply be linear transformations of the data. Consider a 2-layer linear autoencoder as illustrated in Fig. 1. Let the encoder function be described by (9) and the decoder function by (10). If the network is trained using the mean squared error shown in (11), as the loss function J, the minimum reconstruction error is shown to be achieved if $\mathbf{V} = \mathbf{W}$ and $\mathbf{c} = \boldsymbol{\mu}$, where the columns of $\mathbf{W}$ span the orthonormal basis spanned by the eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$ of the dataset (Goodfellow et al., 2016). The columns of $\mathbf{W}$ are ordered by the magnitude of their corresponding eigenvalues. One recognizes that the encoder function $f(\mathbf{x})$ is in fact the same as the KL-transform for the case of a linear autoencoder. This allows for efficient calculations, as the covariance matrix and its corresponding eigenvectors and eigenvalues can easily be calculated, significantly saving computation time compared to training a network. The eigenvectors calculated here capture the directions in which there is the greatest degree of variation in the data. Data can, therefore, be compressed and reconstructed as a linear combination of the projections of that data onto a subspace spanned by the top l eigenvectors with the largest eigenvalues.

$$\mathbf{h} = f(\mathbf{x}) = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}) \tag{9}$$

$$\hat{\mathbf{x}} = g(\mathbf{h}) = \mathbf{V}\mathbf{h} + \mathbf{c} \tag{10}$$

$$J = E\left[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right] \tag{11}$$

The basis of the method is to train two linear autoencoders. All forward trajectories belonging to the class of the selected vessel are input to the forward linear autoencoder. In the latent representation, i.e. the code space, one can then interpolate between existing data points, and in this manner predict the latent representation of the selected vessel's future trajectory. If one then runs a forward pass through the decoder, i.e. (10), one will get a full trajectory prediction, at the cost of a matrix multiplication operation. One can in theory move about the latent space and generate new trajectories in a similar manner to the MNIST digits in Kingma and Welling (2014). The underlying distribution of possible future trajectories would then be visualized, where moving in one dimension or another represents the most variation in the possible future trajectories. The interpolation, however, depends on a similarity measure of the backward trajectories to the backward, i.e. observed, trajectory of the selected vessel. This is facilitated via the backward linear autoencoder.

2.2.1. Forward linear autoencoder

The forward linear autoencoder has as its goal to create a meaningful latent representation of the extracted forward trajectories. However, training an autoencoder on all the forward trajectories will yield a latent representation that describes the greatest variations in the data, i.e. between all clusters of trajectories. This may for instance yield predictions where a data point is interpolated between clusters, and in fact represents an unrealistic data point that is not part of the original distribution. If one, however, considers solely the cluster of trajectories that the selected vessel has been classified to, one now has a subset of trajectories that are highly similar to each other, where interpolation between points should be meaningful. As such, training an autoencoder on this subset of data will allow it to learn a latent representation that describes this specific cluster. Decoding a data point from this latent representation will, therefore, yield a trajectory prediction of higher fidelity. The encoder and decoder functions are shown in (12) and (13) respectively, where $\mathbf{E}_f$ is the matrix of the subset of the top l eigenvectors of the covariance matrix of the forward trajectories.

$$\mathbf{h}_f = \mathbf{E}_f^T \mathbf{f} \tag{12}$$

$$\hat{\mathbf{f}} = \mathbf{E}_f \mathbf{h}_f \tag{13}$$

2.2.2. Backward linear autoencoder

The success of the trajectory prediction technique relies on the interpolation in the latent space of the forward linear autoencoder. Given that the future trajectory is unknown, one must infer the latent representation of the selected vessel in the forward latent representation. It is, therefore, suggested to investigate the backward trajectories of the classified cluster in comparison to the backward trajectory of the selected vessel. By identifying the degree of similarity between all backward trajectories in the cluster, and the backward trajectory of the selected vessel, one can interpolate in the latent space of the forward linear autoencoder, using the similarity of the backward trajectories as weights.

It is suggested in this study to utilize a linear autoencoder to evaluate the similarity. In the same manner as the forward linear autoencoder, the backward linear autoencoder will learn a meaningful latent representation that describes the variation in the underlying trajectory data. In this lower dimensional latent space, the distance from the encoded selected vessel trajectory to all other trajectories can be measured. Conducting such a similarity measure in this space will yield better results due to the same challenges relating to curse of dimensionality (Steinbach et al., 2004) as those addressed in 2.1.2. The encoder and decoder functions are shown in (14) and (15) respectively, where $\mathbf{E}_b$ is the matrix of the subset of top l eigenvectors of the covariance matrix of the backward trajectories.

$$\mathbf{h}_b = \mathbf{E}_b^T \mathbf{b} \tag{14}$$

$$\hat{\mathbf{b}} = \mathbf{E}_b \mathbf{h}_b \tag{15}$$

2.2.3. Latent interpolation

Since there is no explicit mapping function from the latent space of the backward autoencoder to the latent space of the forward autoencoder, a similarity-based mapping approach is suggested. The architecture of the suggested method is visualized in Fig. 4. The figure shows how the backward trajectories are mapped to a latent representation, $\mathbf{h}_b$, in orange, as are the forward trajectories in green to the latent space $\mathbf{h}_f$. $\mathbf{e}$ represents the coordinate systems of the latent spaces. The backward trajectory of the selected vessel is illustrated as the solid red line, and represents the information available of the past behavior of the selected vessel. This information is then encoded in the backward latent representation, $\mathbf{h}_{b,s}$, as the red data point. The goal of the mapping operation is to map to the corresponding red data point in the latent representation of the forward trajectories.

The mapping function can be considered an interpolation between the data points of the encoded forward trajectories. The similarity between the encoded backward trajectory of the selected vessel $\mathbf{h}_{b,s}$, and all the backward trajectories is calculated as the Euclidean distance according to (16). One common form of interpolation for multivariate data is inverse distance weighting. An interpolation scheme is presented in Shepard (1968) with a weighting function according to (17), and the interpolated value calculated according to (18). The equation interpolates within a neighborhood, such that the k nearest data points are found, and the interpolated value is calculated on a subset of neighboring data. In this manner, the interpolated value is not as affected by outliers, and, therefore, more likely to be closer to the true value.

$$d_i = \|\mathbf{h}_{b,i} - \mathbf{h}_{b,s}\| \tag{16}$$

$$w_i = \frac{1}{d_i} \tag{17}$$

$$\hat{\mathbf{h}}_{f,s} = \frac{\sum_{i=1}^{k} w_i \mathbf{h}_{f,i}}{\sum_{i=1}^{k} w_i} \tag{18}$$

2.2.4. Decoded trajectory prediction

Subsequent to the latent interpolation operation, the future trajectory of the selected vessel can be decoded, i.e. predicted, according to (19). Once this is completed, $\hat{\mathbf{f}}_s$ must be reshaped to a matrix containing the spatial data (x, y) as its columns. The prediction is subsequently updated such that the offset between the true initial position (x_0, y_0) and the predicted initial position $(\hat{x}_0, \hat{y}_0)$ is subtracted from all the entries of the prediction to account for minor offsets that occur due to the approximation inherent in the latent interpolation. This yields a trajectory prediction for the selected vessel at 30 s intervals, up to the desired prediction horizon, T_p. One can evaluate each row of the matrix as the predicted vessel position, $\hat{\mathbf{p}}_j$, in each vessel state, $\mathbf{s}_j$, where each state is separated by 30 s.

$$\hat{\mathbf{f}}_s = \mathbf{E}_f \hat{\mathbf{h}}_{f,s} \tag{19}$$

2.2.5. Uncertainty estimate of predicted position

The trajectory prediction gives a single prediction. However, the outlined method does not give a measure of uncertainty related to the predicted position at each time interval. A method is therefore suggested to achieve this utilizing the linear autoencoder architecture previously introduced. Some uncertainty can be attributed to the reconstruction loss that results from reducing the dimensionality in the autoencoders, but the primary source is the uncertainty associated with the latent interpolation.

It is, therefore, suggested to create a distribution in the latent space of the encoded forward trajectories, i.e. $\mathbf{H}_{f,s}$, that can account for some of the interpolation error. If one considers the neighborhood of $\hat{\mathbf{h}}_{f,s}$, one can investigate the uncertainty with respect to the k nearest neighbors in the latent representation of the backward trajectories, $\mathbf{h}_b$. The method suggests to assume that $\hat{\mathbf{h}}_{f,s}$ is the mean of a normal distribution according to (20), with a weighted unbiased covariance according to (21). These weights correspond to those in (17). In this manner, the distribution will reflect the relevant importance of each latent forward trajectory representation, based on their weights from the backward trajectory similarity measure.

$$\mathbf{H}_{f,s} \sim \mathcal{N}(\hat{\mathbf{h}}_{f,s}, \boldsymbol{\Sigma}_{h,s}) \tag{20}$$

$$\boldsymbol{\Sigma}_{h,s} = \frac{\sum_{i=1}^{k} w_i (\mathbf{h}_{f,i} - \hat{\mathbf{h}}_{f,s})(\mathbf{h}_{f,i} - \hat{\mathbf{h}}_{f,s})^T}{\sum_{i=1}^{k} w_i - \frac{\sum_{i=1}^{k} w_i^2}{\sum_{i=1}^{k} w_i}} \tag{21}$$

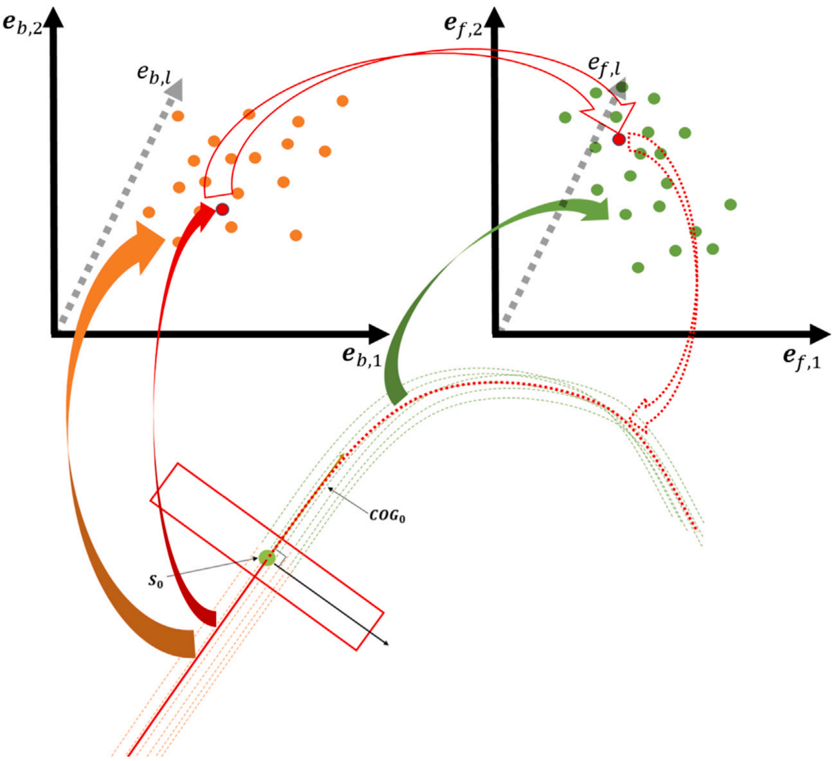

Fig. 4. Illustration of the latent representations in the dual linear autoencoder architecture. The interpolation between the known latent representation of the backward trajectory of the selected vessel to the unknown forward latent representation is also illustrated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

This, however, only yields a form of uncertainty with respect to the latent representation of the selected vessel's forward trajectory. What is of interest, however, is the uncertainty of the predicted position at various vessel states. To achieve this, it is suggested to run a Monte Carlo simulation (Raychaudhuri, 2008) that samples from the latent normal distribution in (20) to approximate the distribution of the trajectory predictions. Each sample from $\mathbf{H}_{f,s}$ is decoded according to (19), yielding a full trajectory prediction. In the same manner as in Section 2.2.4, the sampled predictions are also updated based on the offset of the true and predicted initial positions. This correction will be greater for samples further away from the true value of $\mathbf{h}_{f,s}$, but it is assumed to have limited effect on the predictions with respect to estimating the uncertainty. After the samples are decoded, the distribution of the decoded trajectory positions can be evaluated at each time instance. This can be viewed as the distribution $\mathbf{P}_j$ of the predicted position $\hat{\mathbf{p}}_j$ for each vessel state $\mathbf{s}_j$, where each state is that of the selected vessel at 30 s intervals. The distribution of the position in each state can further be assumed to be normally distributed according to (22), where the mean and covariance are calculated based on the sampled predictions. As such, uncertainty measures can be calculated with respect to the standard deviation of the distribution.

$$\mathbf{P}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{22}$$

3. Results and discussion

To evaluate the method, 100 random data points were selected from a dataset of historical AIS data in the region surrounding the city of Tromsø, Norway. The dataset corresponds to that collected from January 1st, 2017 to January 1st, 2018. Each data point represents a selected vessel state, that will be initialized as the initial state, $\mathbf{s}_0$, of that vessel. The aforementioned trajectory prediction methodology is then utilized to predict the future 30 min of each selected vessel's trajectory, such as to evaluate the performance of the method. The true future trajectories of the selected vessels can be thought of as the test dataset for each respective prediction. The remainder of the AIS data is then the training dataset utilized to conduct the predictions. In this manner, the method predicts the future trajectory of 100 different vessels, and the accuracy can be evaluated based on the true trajectory of the vessel. A value of $l = 3$ is utilized for the latent representation of both $\mathbf{h}_b$ and $\mathbf{h}_f$. Additionally, the 100 most similar vessels to each selected vessel, i.e. $k = 100$, is utilized for the latent interpolation.

3.1. Classification accuracy

The input to the trajectory prediction module is the set of extracted trajectories corresponding to the output of the classification module. As such, the method relies on the accuracy of this classification, as an incorrect classification will result in a prediction with respect to a cluster of ship behavior that does not match the selected vessel. In this study, a value of $k = 7$ was utilized for the kNN classifier. For the results presented in this section, 67% of the selected vessels were classified correctly.

In many cases, however, the incorrect classification can be attributed an incorrect behavior mode, i.e. cluster. These modes can be along the correct route, but may for instance traverse further to one side of the lane, or have variations in the speed profile along the route. Predictions with respect to these modes, despite being incorrect, can nonetheless result in reasonably accurate predictions. This is due to the clustering algorithm identifying multiple modes that are quite similar. One should note that this can be seen as a situation where some data clusters can overlap each other. As such, the 33% of incorrectly classified cases in this study likely includes many cases in which the selected vessel was classified to an incorrect mode along the correct route, i.e. a similar trajectory mode.

Additionally, the success of the classification depends on the complexity of the discovered clusters. It is on the one hand desirable for the clustering algorithm to discover as many trajectory modes as

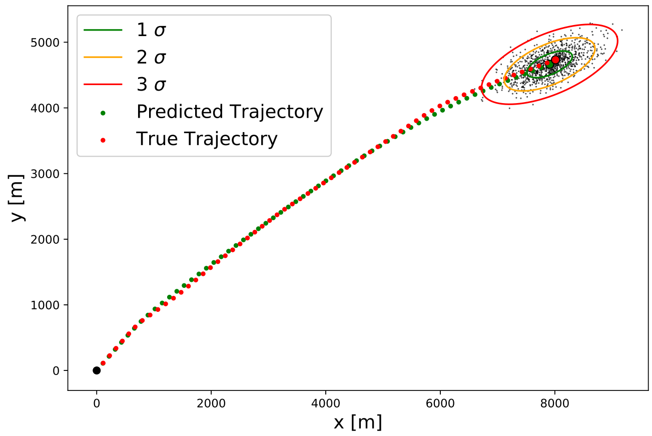

Fig. 5. Trajectory prediction with correct classification. Contours are illustrated for $T_P = 30$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

possible, as this can enhance the accuracy of the subsequent trajectory prediction. On the other hand, an increase in the number of clusters will provide a more difficult classification task. This will be further complicated due to overlapping clusters as mentioned previously. Each cluster is multi-dimensional, and classifying in this space can be challenging. However, the focus of the study is the dual linear autoencoder prediction technique.

It should be noted that the trajectory prediction methodology utilizing a dual linear autoencoder as described in this study can be utilized based on any previous clustering and classification technique. However, it does require that trajectories are extracted utilizing the methods described in Section 2.1.1, such that trajectories can be encoded and decoded properly. This trajectory extraction process can, however, also be conducted after an alternative clustering and classification regime has been utilized.

3.2. Trajectory prediction

Fig. 5 illustrates an example of a trajectory prediction for one of the randomly selected vessels in the dataset. All presented predictions are evaluated with $\mathbf{p}_0$ as the origin of the coordinate system to more easily evaluate the distances involved. In the case of Fig. 5, the algorithm classified the selected vessel to the correct cluster of trajectories. The green dotted line represents the predicted trajectory of the selected vessel, and the red dotted line represents the true trajectory of the selected vessel. It is clear that for this case, the trajectory prediction was quite accurate. Additionally, an estimate of the uncertainty of the position at a 30 min prediction horizon is illustrated. Each black dot illustrated represents a decoded sample from the normal distribution of $\mathbf{H}_{f,s}$ in (20). Utilizing these predictions, a normal distribution $\mathbf{P}_j$, was fit to the predicted positions for each state according to (22). Based on this distribution, the 1σ, 2σ and 3σ contours could be evaluated, and are visualized in the figure. Such contours can be evaluated at any prediction horizon, but only the results for the 30 min prediction horizons are illustrated in this section. Fig. 6 shows a prediction of a more complex trajectory, showing that the method is also able to successfully reconstruct more complex trajectories.

In certain cases, however, the classification is incorrect. This, in some cases, can result in a vessel prediction along an incorrect route, resulting in a degree of error with respect to the predicted position. An example of such a case is illustrated in Fig. 7.

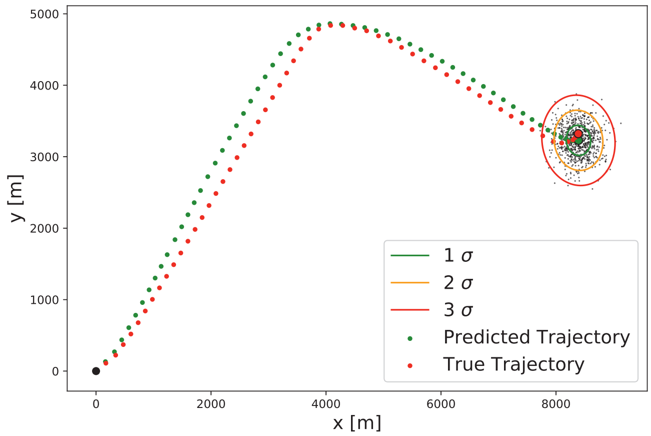

Fig. 6. Prediction of complex trajectory. Contours are illustrated for $T_P = 30$.

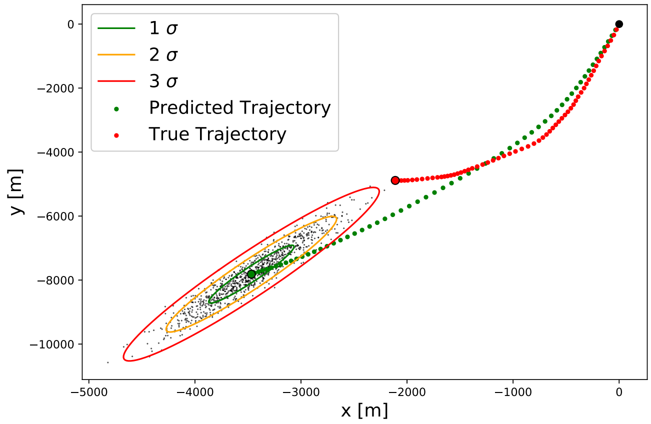

Fig. 7. Trajectory prediction with incorrect classification. Contours are illustrated for $T_P = 30$.

3.3. Prediction accuracy

The performance of the method is most effectively measured based on the accuracy of the prediction with respect to the true vessel position. The predicted position error is calculated as the distance from the mean, $\boldsymbol{\mu}_j$, of the distribution $\mathbf{P}_j$, and the true position in that state, i.e. $\mathbf{p}_j$. Additionally, the error is presented as a percentage of the true distance traveled by the selected vessel. This is due to various vessels having traveled different distances during the course of 30 min. In this manner, one can compare the error irrespective of the distance traveled.

Fig. 8 illustrates the median error of all 100 predictions as a function of time, i.e. the desired prediction horizon. The overall error for all selected vessels is evaluated, where the median error for a 30 min prediction is found to be 2.5%. If one looks only at the vessels that were incorrectly classified, it is evident that they result in a higher degree of error, where the median error at a prediction horizon of 30 min is 9.6% of the distance traveled. If one solely investigates the correctly classified vessels, however, the accuracy of the prediction increases significantly, with a median position error of 1.6% for a prediction horizon of 30 min. This illustrates the importance of correctly classifying the selected vessel, as the predictions are discrete with respect to each class of trajectories. An incorrect classification results in the prediction being conducted on a cluster corresponding to a different mode of ship behavior than that of the selected vessel. Fig. 9 illustrates a box plot of the positional error at 5 min intervals for the correctly classified vessels. The green bars correspond to the median values in Fig. 8. It is clear that the variance of the error increases as a function of

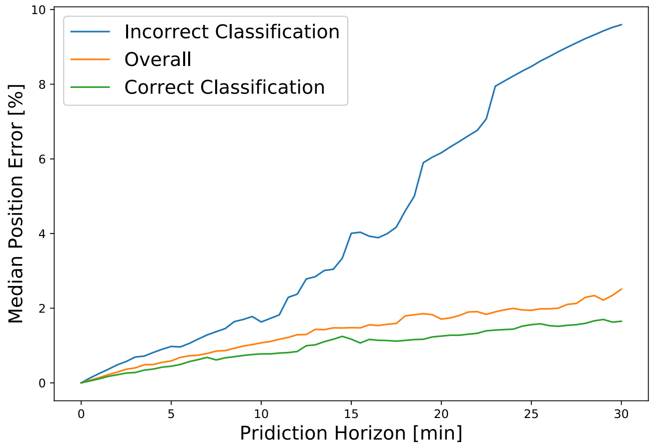

Fig. 8. Median position error of 100 randomly selected vessels as a function of the prediction horizon. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

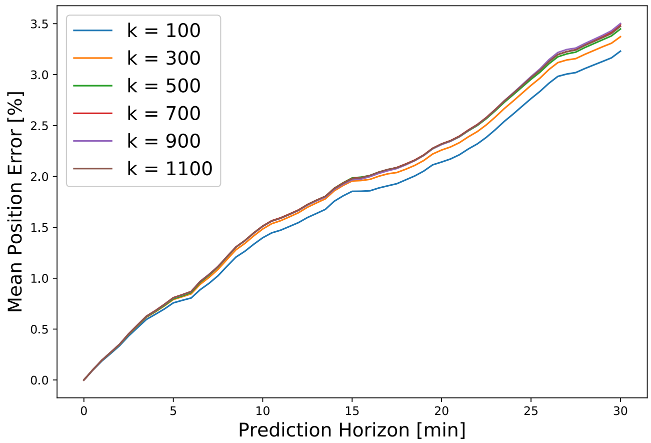

Fig. 10. Mean error for values of k used in interpolation.

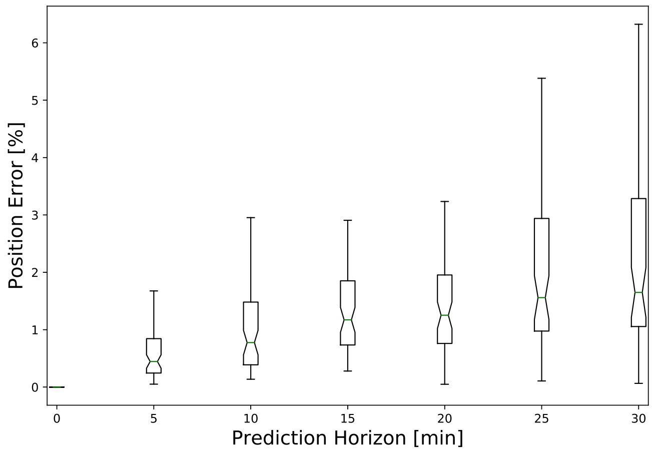

Fig. 9. Box plot of error for correctly classified vessels.

time. Generally, it appears that the method has good performance when vessels are correctly classified for prediction horizons up to 30 min.

3.4. Position uncertainty estimate

As previously mentioned, a normal distribution of the predicted position $\mathbf{P}_j$ in each state is evaluated according to (22). The resultant 1σ, 2σ and 3σ contours can be utilized to give a measure of uncertainty relating to the prediction in each state. It is desirable for this uncertainty to be as small as possible whilst still capturing the behavior of the selected vessel. If the uncertainty is too small, however, it may not include the true future position of the selected vessel. Allowing for too much uncertainty on the other hand, is not desirable either. It is conceivable to extend the region of uncertainty such that the true future position of the selected vessel will always be included within the contours. In such cases, however, the usefulness with respect to maritime situation awareness will be degraded, as there will be a risk of collision for a very large area. This increases the likelihood of the navigator needing to take action in cases where the true risk of collision is low.

The degree of uncertainty is dependent upon two factors: the power of the clustering algorithm, and the number of vessels included in the latent interpolation. If the clustering algorithm is able to discover a group of trajectories with very specific behavior, the uncertainty of the prediction will decrease as the variance of the behavior within the cluster is limited. In essence, all the historical trajectories utilized in the prediction are almost the same, causing all predictions to have similar values when sampling from the latent space. Similarly, if one only selects a limited number of vessels to evaluate the latent representation of the selected vessel in (20), the prediction will be restricted to the behavior of these historical trajectories. Increasing the number of trajectories will increase the variance of the behavior, and contribute to a larger region of uncertainty. The effect of this was investigated, where the number of similar vessels, k, utilized in (20), was varied and the position error evaluated as a percentage of the distance traveled in the same manner as described in Section 3.3. Fig. 10 illustrates the mean error for various values of k as a function of the prediction horizon for the 100 randomly selected vessels. It is evident that increasing the value of k contributes to an increase in error, but that this error converges as k increases.

It should be noted that the probability contours solely relate to the probability of the predicted future position, and as such are entirely dependent upon the model developed in this study. They provide a measure of uncertainty with respect to the predicted positions, where the true position should fall within the region enclosed by the contours. The predictions are based on historically similar vessels, whose behavior do not necessarily match that of the selected vessel. The assumption that the future trajectory of a selected vessel depends on its past trajectory is, therefore, a limitation of the method. The uncertainty of the predictions can, therefore, be thought of as describing the variance of the historical behavior, where it is likely that the vessel will fall somewhere within the specified region. The method is designed to identify the most similar trajectories. However, the data may be dominated by specific vessel behavior that has a higher frequency. If more similar historical trajectories have a lower frequency, the data will be dominated by the less similar trajectories of the highest frequency. This effect will, however, be somewhat ameliorated due to the weights in (17), as the more similar trajectories will have higher weights when calculating $\hat{\mathbf{h}}_{f,s}$ and $\boldsymbol{\Sigma}_{h,s}$.

The percentage of the correctly classified vessels whose true position after 30 min was within the regions bounded by the corresponding σ-contours was also investigated. This was conducted in order to evaluate the uncertainty measure's ability to capture the true position of the selected vessel. The results are shown in Table 1. The values are estimated such that the 3σ contour includes the points inside the 2σ contour which again includes the points inside the 1σ contour. 75% of all true vessel positions were captured by the 3σ contour for the tested selected vessels. Cases in which the true positions did not reside within the contours were, therefore, investigated.

Fig. 11 illustrates one such case. The prediction appears to be quite accurate, with the true and predicted trajectories nearly exactly

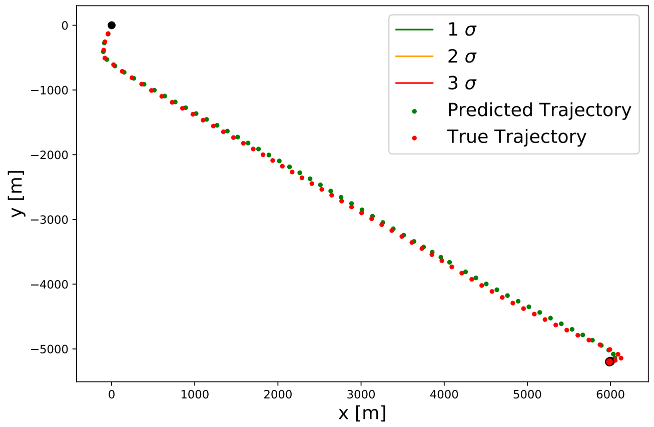

Fig. 11. Prediction of ferry trajectory. $T_P = 30$.

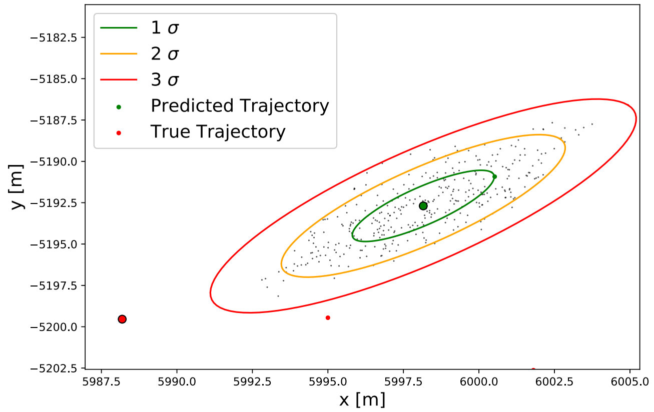

Fig. 12. Uncertainty contours for ferry trajectory prediction.

aligned. It is evident, however, that the uncertainty contours are not visible. Fig. 12 illustrates a close up of the predicted position after 30 min. Here one can see that the predicted final position, illustrated as the largest red dot, falls outside the uncertainty contours. When investigating the scale involved, one can see that the true and predicted positions in fact reside less than 20 m from each other. The uncertainty ellipses are extremely small, and therefore reflect a very high certainty of the model with respect to the predicted position. Fig. 13 illustrates the cluster of trajectories utilized to conduct the prediction. It is discovered that this cluster corresponds to a ferry, where the final position after 30 min is at one of its ports. As such, the data upon which the prediction is determined is concentrated about this position. An offset of 20 m can be accounted for by error inherent in the AIS data, in addition to the orientation of the vessel when in a port.

There are multiple ferries in the region surrounding Tromsø. When investigating all tested vessels with a ship type of "Passenger Vessel", it was found that the performance with respect to the uncertainty measures was degraded, despite the predictions being quite accurate. The percentage of these vessels is also shown in Table 1, in addition to results for all vessels except those labeled as "Passenger Vessel". It is clear that the performance increases in this case. The performance of other vessel types is, however, likely affected by vessels with similar effects, where the cluster of underlying data is too similar to allow for a large uncertainty measure, whilst still providing accurate predictions.

3.5. Running time

As the algorithm is intended to run live, it is of interest to investigate the running time of the method. To support the discussion, the authors have evaluated the running time of the method for the cases in this study. All evaluations have been run on a 2.30 GHz CPU and 16 GB ram. In the following sections, the running times are evaluated in two parts. The first addresses the running time of the trajectory extraction algorithm, and the other the classification, clustering and prediction algorithms.

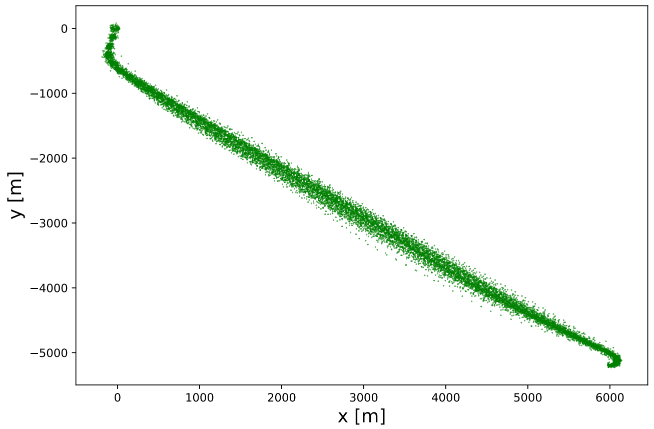

Fig. 13. Cluster used for ferry trajectory prediction.

Table 1
Percentage of true 30 min selected vessel position inside corresponding σ-contour.

σ	All vessels	Passenger vessels	Without passenger vessels
1	37%	11%	41%
2	63%	22%	70%
3	75%	33%	82%

The data utilized in this study consisted of approximately 15 million data points. These data points are input to the algorithm without any pre-processing. In the trajectory extraction step, relevant historical trajectories need to be extracted from the raw AIS data. Once this is conducted, the data will be available for the period a prediction is required. Fig. 14 illustrates the running time of the extraction of trajectory data. This is visualized as a function of the number of relevant trajectories extracted. A third order regression was applied to the data to visualize the relationship between the running time and the number of trajectories. It is clear that the running time increases with the number of extracted trajectories. It appears that for most cases, the trajectories were extracted within two minutes. The trajectory extraction process only needs to be conducted once, and subsequent predictions can utilize the previously extracted data. The algorithm utilized to extract the trajectories from the raw data has not been optimized in the current implementation, however. As such, the running time of the extraction phase can likely be significantly improved through optimization. Additionally, in a future system utilized for vessel trajectory prediction, a more advanced computer would be utilized to conduct the prediction. Furthermore, speed can be increased by pre-processing data for regions such that whole trajectories are available for extraction, instead of raw data points that require trajectories to be created. Nonetheless, the extraction times evaluated in the implementation in this study are reasonable for the outlined purposes.

Of most interest to the study is arguably the performance of the clustering, classification and prediction algorithms. Fig. 15 illustrates the individual algorithm running times in addition to the total running time, i.e. the sum of the clustering, classification and prediction running times. These are again plotted as a function of the number of extracted trajectories with a third order regression. It is clear that all algorithms are quite fast. The classification was virtually instantaneous for all cases, and the dual linear autoencoder trajectory prediction took less than one second for all cases. The clustering algorithm dominates

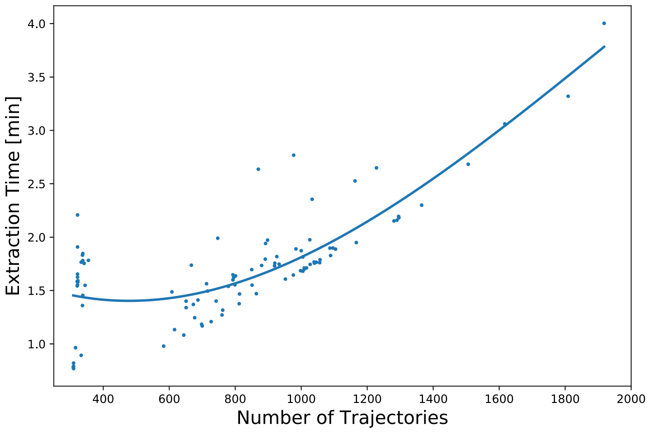

Fig. 14. Trajectory extraction running time.

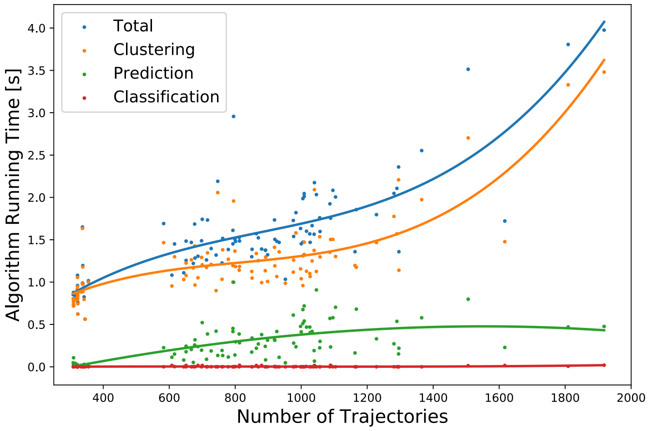

Fig. 15. Algorithm running times.

the total running time, where most cases took between one and two seconds. However, the overall total running time for all algorithms was nonetheless quite low, with the worst case being just below four seconds, and the majority of the evaluated cases below two seconds. This is considered to be acceptable for the purposes of this study. With a more advanced computer, and optimized implementation, the running time would likely be even lower.

4. Conclusion

A linear version of the autoencoder is implemented in this study, and it is shown that it can predict complex trajectories with a high degree of accuracy. Training the linear version of the autoencoder utilized in this study is also less computationally demanding than deeper autoencoders. Compared to methods that predict future states conditioned upon their prediction of the previous state, this method draws upon the generative ability of autoencoders to predict entire trajectories. Generative models have been shown to have good performance in creating new data points that belong to the distribution of the training data. By interpolating in the latent space of historical trajectories, the method in this study is able generate an entirely new trajectory. The method is, however, dependent on the ability to cluster the trajectories. If one were to apply the same method to all historical trajectories, as opposed to a cluster, one may end up interpolating between clusters. As such, a subsequent prediction will result in an unrealistic trajectory that does not belong to the distribution of the historical data.

Applying the method on a single cluster, however, will increase the ability to describe subtle differences between trajectories, thereby enhancing the subsequent prediction. Also, given that all trajectories within the same cluster are quite similar, the likelihood of generating a trajectory that does not belong to the original distribution is unlikely. Additionally, as the method generates an entire trajectory, and not iterative states conditioned upon the previous prediction, prediction errors will not propagate as a function of time. The error will, therefore, be related to the error of the entire trajectory. By evaluating multiple trajectory predictions, however, one can estimate the degree of uncertainty of the prediction, and this uncertainty can be modeled using the outlined method in this study.

The approach suggested in this study provides an effective method to predict the future trajectories of ocean going vessels. Specifically, the method provides the basis for an advanced ship predictor on a global scale. This ship predictor will aid in providing situation awareness to navigators, in that the future trajectory of potential target vessels can be predicted far in advance. Based on a subsequent evaluation of the collision risk, simple corrective measures can be conducted to prevent close-range encounter situations from arising. If effective, such a method will increase the safety associated with maritime operations. Such situation awareness can also potentially be extended to autonomous vessels, which can make system level intelligent decisions based on input from the outlined approach.

Future work will include investigating deep learning methodologies that introduce nonlinearity, and how increases in the complexity of the model can potentially increase the performance of the predictions. In addition, further work will be conducted on integrating such methods into an advanced ship predictor to provide situation awareness to navigators.

CRediT authorship contribution statement

Brian Murray: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Visualization. **Lokukaluge Prasad Perera:** Conceptualization, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Norwegian Ministry of Education and Research and the MARKOM-2020 project, a development project for maritime competence established by the Norwegian Ministry of Education and Research in cooperation with the Norwegian Ministry of Trade and Industry. The authors would also like to express gratitude to the Norwegian Coastal Administration for providing access to their AIS database.

References

Aarsæther, K.G., Moan, T., 2009. Estimating navigation patterns from AIS. J. Navig. 62 (04), 587–607. http://dx.doi.org/10.1017/S0373463309990129.

Bourlard, H., Kamp, Y., 1988. Auto-association by multilayer perceptrons and singular value decomposition. Biol. Cybernet. 59 (4–5), 291–294. http://dx.doi.org/10.1007/BF00332918.

Dalsnes, B.R., Hexeberg, S., Flåten, A.L., Eriksen, B.-O.H., Brekke, E.F., 2018. The neighbor course distribution method with Gaussian mixture models for AIS-based vessel trajectory prediction. In: 2018 21st International Conference on Information Fusion. FUSION, IEEE, pp. 580–587.

Endsley, M.R., 1995. Toward a theory of situation awareness in dynamic systems. Hum. Factors: J. Hum. Factors Ergon. Soc. 37 (1), 32–64. http://dx.doi.org/10.1518/001872095779049543.

Endsley, M.R., Bolté, B., Jones, D.G., 2003. Designing for Situation Awareness : An Approach to User-Centered Design. Taylor & Francis.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugen. 7 (2), 179–188. http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x.

Foster, D., 2019. Generative Deep Learning. O'Reilly Media, Inc..

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Hermann, M., Pentek, T., Otto, B., 2016. Design principles for industrie 4.0 scenarios. In: 2016 49th Hawaii International Conference on System Sciences. HICSS, IEEE, pp. 3928–3937. http://dx.doi.org/10.1109/HICSS.2016.488.

Hexeberg, S., Flaten, A.L., Eriksen, B.-O.H., Brekke, E.F., 2017. AIS-based vessel trajectory prediction. In: 2017 20th International Conference on Information Fusion. FUSION, IEEE, http://dx.doi.org/10.23919/ICIF.2017.8009762.

Karhunen, K., 1946. Zur spektraltheorie stochastischer prozesse. Ann. Acad. Sci. Fenn. 37.

Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes. In: Proceedings of the International Conference on Learning Representations. ICLR, http://arxiv.org/abs/1312.6114.

Mazzarella, F., Arguedas, V.F., Vespe, M., 2015. Knowledge-based vessel position prediction using historical AIS data. In: 2015 Sensor Data Fusion: Trends, Solutions, Applications. SDF, IEEE, pp. 1–6. http://dx.doi.org/10.1109/SDF.2015.7347707.

Murray, B., Perera, L.P., 2019. An AIS-based multiple trajectory prediction approach for collision avoidance in future vessels. In: 38th International Conference on Ocean, Offshore & Arctic Engineering, Vol. 7B-2019. OMAE 2019, American Society of Mechanical Engineers (ASME), http://dx.doi.org/10.1115/OMAE2019-95963.

Pallotta, G., Horn, S., Braca, P., Bryan, K., 2014. Context-enhanced vessel prediction based on ornstein-uhlenbeck processes using historical AIS traffic patterns : Real-world experimental results. In: Information Fusion (FUSION), 2014 17th International Conference on. July, pp. 1–7.

Pallotta, G., Vespe, M., Bryan, K., 2013. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. Entropy 15 (12), 2218–2245. http://dx.doi.org/10.3390/e15062218.

Perera, L.P., 2017. Navigation vector based ship maneuvering prediction. Ocean Eng. 138, 151–160. http://dx.doi.org/10.1016/j.oceaneng.2017.04.017.

Perera, L.P., Carvalho, J.P., Guedes Soares, C., 2010. Autonomous guidance and navigation based on the COLREGs rules and regulations of collision avoidance. Adv. Ship Des. Pollut. Prev. 205–216. http://dx.doi.org/10.1201/b10565-26.

Perera, L.P., Guedes Soares, C., 2015. Collision risk detection and quantification in ship navigation with integrated bridge systems. Ocean Eng. 109, 344–354. http://dx.doi.org/10.1016/j.oceaneng.2015.08.016.

Perera, L.P., Mo, B., 2018. Ship performance and navigation data compression and communication under autoencoder system architecture. J. Ocean Eng. Sci. 3 (2), 133–143. http://dx.doi.org/10.1016/j.joes.2018.04.002.

Perera, L.P., Murray, B., 2019. Situation awareness of autonomous ship navigation in a mixed environment under advanced ship predictor. In: 38th International Conference on Ocean, Offshore & Arctic Engineering, Vol. 7B-2019. OMAE 2019, American Society of Mechanical Engineers (ASME), http://dx.doi.org/10.1115/OMAE2019-95571.

Perera, L.P., Oliveira, P., Guedes Soares, C., 2012. Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. IEEE Trans. Intell. Transp. Syst. 13 (3), 1188–1200. http://dx.doi.org/10.1109/TITS.2012.2187282.

Raychaudhuri, S., 2008. Introduction to Monte Carlo simulation. In: 2008 Winter Simulation Conference. IEEE, pp. 91–100. http://dx.doi.org/10.1109/WSC.2008.4736059.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted Gaussian mixture models. Digit. Signal Process. 10 (1–3), 19–41.

Rezende, D.J., Mohamed, S., Wierstra, D., 2014. Stochastic backpropagation and approximate inference in deep generative models. In: Xing, E.P., Jebara, T. (Eds.), Proceedings of the 31st International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 32, (2), PMLR, Bejing, China, pp. 1278–1286, http://proceedings.mlr.press/v32/rezende14.html.

Ristic, B., Scala, B.L., Morelande, M., Gordon, N., 2008. Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. In: 2008 11th International Conference on Information Fusion. pp. 40–46. http://dx.doi.org/10.1109/ICIF.2008.4632190.

Rødseth, Ø.J., Perera, L.P., Mo, B., 2015. Big data in shipping - challenges and opportunities. In: 15th International Conference on Computer and IT Applications in the Maritime Industries. COMPIT '16, pp. 361–373.

Rong, H., Teixeira, A., Guedes Soares, C., 2019. Ship trajectory uncertainty prediction based on a Gaussian process model. Ocean Eng. 182, 499–511. http://dx.doi.org/10.1016/J.OCEANENG.2019.04.024.

Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM National Conference. ACM '68, ACM, New York, NY, USA, pp. 517–524. http://dx.doi.org/10.1145/800186.810616.

Steinbach, M., Ertöz, L., Kumar, V., 2004. The challenges of clustering high dimensional data. In: New Directions in Statistical Physics. Springer, Berlin, Heidelberg, pp. 273–309.

Tam, C., Bucknall, R., Greig, A., 2009. Review of collision avoidance and path planning methods for ships in close range encounters. J. Navig. 62 (03), 455. http://dx.doi.org/10.1017/S0373463308005134.

Tu, E., Zhang, G., Rachmawati, L., Rajabally, E., Huang, G.-B., 2017. Exploiting AIS data for intelligent maritime navigation: A comprehensive survey from data to methodology. IEEE Trans. Intell. Transp. Syst. 1–24. http://dx.doi.org/10.1109/TITS.2017.2724551.

Zhang, L., Meng, Q., 2019. Probabilistic ship domain with applications to ship collision risk assessment. Ocean Eng. 186, http://dx.doi.org/10.1016/j.oceaneng.2019.106130.

Zhang, S.-k., Shi, G.-y., Liu, Z.-j., Zhao, Z.-w., Wu, Z.-l., 2018. Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity. Ocean Eng. 155, 240–250. http://dx.doi.org/10.1016/j.oceaneng.2018.02.060.

Zhao, L., Shi, G., 2019. A trajectory clustering method based on Douglas–Peucker compression and density for marine traffic pattern recognition. Ocean Eng. 456–467. http://dx.doi.org/10.1016/j.oceaneng.2018.12.019.

Zhou, Y., Daamen, W., Vellinga, T., Hoogendoorn, S.P., 2019. Ship classification based on ship behavior clustering from AIS data. Ocean Eng. 176–187. http://dx.doi.org/10.1016/j.oceaneng.2019.02.005.