



UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Technology and Safety

Machine Learning for Enhanced Maritime Situation Awareness

Leveraging Historical AIS Data for Ship Trajectory Prediction

Brian Murray

A dissertation for the degree of Philosophiae Doctor

January 2021



Machine Learning for Enhanced Maritime Situation Awareness

Leveraging Historical AIS Data for Ship Trajectory Prediction

Brian Murray

Doctoral thesis in partial fulfillment of the requirements for
the degree of Philosophiae Doctor

January, 2021

UiT The Arctic University of Norway
Faculty of Science and Technology
Department of Technology and Safety



UiT The Arctic
University of Norway

Abstract

In this thesis, methods to support high level situation awareness in ship navigators through appropriate automation are investigated. Situation awareness relates to the perception of the environment (level 1), comprehension of the situation (level 2), and projection of future dynamics (level 3). Ship navigators likely conduct mental simulations of future ship traffic (level 3 projections), that facilitate proactive collision avoidance actions. Such actions may include minor speed and/or heading alterations that can prevent future close-encounter situations from arising, enhancing the overall safety of maritime operations.

Currently, there is limited automation support for level 3 projections, where the most common approaches utilize linear predictions based on constant speed and course values. Such approaches, however, are not capable of predicting more complex ship behavior. Ship navigators likely facilitate such predictions by developing models for level 3 situation awareness through experience. It is, therefore, suggested in this thesis to develop methods that emulate the development of high level human situation awareness. This is facilitated by leveraging machine learning, where navigational experience is artificially represented by historical AIS data.

First, methods are developed to emulate human situation awareness by developing categorization functions. In this manner, historical ship behavior is categorized to reflect distinct patterns. To facilitate this, machine learning is leveraged to generate meaningful representations of historical AIS trajectories, and discover clusters of specific behavior. Second, methods are developed to facilitate pattern matching of an observed trajectory segment to clusters of historical ship behavior. Finally, the research in this thesis presents methods to predict future ship behavior with respect to a given cluster. Such predictions are, furthermore, on a scale intended to support proactive collision avoidance actions.

Two main approaches are used to facilitate these functions. The first utilizes eigendecomposition-based approaches via locally extracted AIS trajectory segments. Anomaly detection is also facilitated via this approach in support of the outlined functions. The second utilizes deep learning-based approaches applied to regionally extracted trajectories. Both approaches are found to be successful in discovering clusters of specific ship behavior in relevant data sets, classifying a trajectory segment to a given cluster or clusters, as well as predicting the future behavior. Furthermore, the local ship behavior techniques can be trained to facilitate live predictions. The deep learning-based techniques, however, require significantly more training time. These models will, therefore, need to be pre-trained. Once trained, however, the deep learning models will facilitate almost instantaneous predictions.

Acknowledgments

This thesis brings to an end an exciting journey that started in 2017. I consider myself fortunate to have been granted the opportunity to conduct research on such interesting topics. Thanks to the MARKOM2020 project for supporting the joint PhD program in Nautical Operations, as well as the Norwegian Coastal Administration for providing access to their AIS database.

Above all, I would like to thank my main supervisor Associate Professor Lokukaluge Prasad Perera, for the innumerable hours he has spend discussing my research, and guiding me along the path towards my PhD. His expertise and support have been invaluable during this journey. I would also like to thank my co-supervisor, Professor Egil Pedersen, for all our excellent discussions, and for always being there to support me when in need. Thanks also to my co-supervisor Associate Professor Henrique Gaspar for our discussions, and helping to guide me in the right direction.

The support of the nautical science team at UiT has also been invaluable. Special thanks to Associate Professor Bjørn-Morten Batalden, Magne-Petter Sollid and Kåre Johansen. I would also like to thank Associate Professor Karl Gunnar Aarsæther, whose expertise on AIS data I enlisted while he was still at SINTEF.

Thanks also to the Machine Learning Group at UiT for their assistance during the course of my PhD. Special thanks to Associate Professor Michael Kampffmeyer, who spurred my interest in deep learning. I would also like to thank Thomas Johansen for the fruitful conversations that helped to aim me in the right direction. Thanks also to Associate Professor Stian Normann Anfinssen for our talks.

I would also like to express gratitude to my friends and colleagues at the Department of Technology and Safety. Per, you have made our days in the office much more fun. Thank you for your friendship and support over the course of these past years. Bjarte and Masoud, thank you for our lunches, chats and friendship. Thanks also to Khanh, Lise, Yufei, Gunn-Helene, Marit, Lisbeth, Yngve and many more.

I would like to thank my parents, Robert and Unn, for their love and support. I would also like to thank my siblings Katrina, Evan and Erik, as well as my grandfather Stein. Finally, I would like to thank my girlfriend Lise for her love and support through these years.

Brian Murray

Tromsø, Norway

January, 2021

Contents

Abstract	i
Acknowledgments	iii
List of Figures	ix
List of Tables	xi
Abbreviations	xiii
Nomenclature	xv
1 Introduction	1
1.1 Motivation and Background	1
1.2 Research Objectives	4
1.3 Research Contributions	6
1.4 Appended Papers	6
1.5 Outline of The Thesis	8
I Methodology and Context	9
2 Maritime Situation Awareness	11
2.1 Theory of Situation Awareness	12
2.1.1 Mental Modeling	13
2.1.2 Autonomy	17
2.2 Situation Awareness in Ship Navigation	19
2.2.1 Level 1 Situation Awareness	20
2.2.2 Level 2 Situation Awareness	22
2.2.3 Level 3 Situation Awareness	23
2.3 Proactive Collision Avoidance	24
2.3.1 Vessel Encounter Situation	24

2.3.2	Long-Range Trajectory Prediction	28
2.4	Ship Behavior Prediction	31
3	Machine Learning	33
3.1	Machine Learning for Enhanced Maritime Situation Awareness . .	33
3.2	Historical AIS Data	35
3.3	Eigendecomposition-Based Dimensionality Reduction	37
3.3.1	Principle Component Analysis	37
3.3.2	Linear Discriminant Analysis	38
3.4	Clustering	39
3.4.1	Gaussian Mixture Models	40
3.4.2	Hierarchical Density-Based Clustering of Applications with Noise	42
3.5	Anomaly Detection	43
3.6	Deep Learning	44
3.6.1	Multi-Layer Perceptron	44
3.6.2	Recurrent Neural Networks	46
3.6.3	Autoencoders	49
II	Research Outcome	55
4	Summary of Research	57
4.1	Local Ship Behavior Prediction	57
4.2	Regional Ship Behavior Prediction	64
4.3	Research Contributions	67
5	Discussion	69
5.1	Contributions to Research Objectives	69
5.2	General Discussion	71
5.2.1	Level 3 Situation Awareness Support	71
5.2.2	Historical AIS Data	73
5.2.3	Machine Learning and Human Situation Awareness	74
5.2.4	Possible Applications	76
6	Concluding Remarks	77
6.1	Conclusions	77
6.2	Suggestions for Further Work	78

Bibliography	81
III Appended Papers	91
Paper I	93
Paper II	111
Paper III	127
Paper IV	139
Paper V	149

List of Figures

2.1	Model of situation awareness adapted from Endsley (1995).	12
2.2	Development of mental models.	16
2.3	Role of mental models adapted from Endsley & Garland (2000b).	17
2.4	Flow chart of collision risk evaluation adapted from Tam & Bucknall (2010). Collision risk is evaluated from the perspective of the own ship with respect to a target ship.	26
2.5	Collision situation stages adapted from Cockcroft & Lameijer (2011). A crossing situation in open seas is used for illustration, with permissible actions by the stand-on vessel.	27
2.6	Examples of encounter situations.	29
3.1	Perceptron.	45
3.2	Multi-Layer Perceptron. Each neuron is a perceptron.	45
3.3	RNN. Figure from Paper V.	47
3.4	P -layer stacked RNN. Each layer is denoted l . Figure from Paper V.	47
3.5	Gated Recurrent Unit architecture. Figure from Paper IV.	48
3.6	Linear autoencoder. The encoder is illustrated in green, and the decoder in orange. Figure adapted from Paper II.	50
3.7	Recurrent autoencoder. The encoder is illustrated in green, and the decoder in orange. Figure from Paper IV.	51
3.8	Variational recurrent autoencoder. The encoder is illustrated in green, and the decoder in orange. Figure adapted from Paper IV.	53
4.1	The extracted trajectories (top left) and clusters of trajectories (top right), along with the classified cluster (bottom left) and predicted future trajectory (bottom right). Figures from Paper I.	59
4.2	Examples of encounter situations.	61
4.3	Intermediate anomaly detection and removal. Figures from Paper III.	63
4.4	Regional trajectory clustering. Figures from Paper IV.	65
4.5	Relevant trajectory clusters (left), and resultant predictions (right) for a test case. Figures from Paper V.	66

List of Tables

3.1	AIS data.	36
4.1	Paper contributions.	67
5.1	Contributions to research objectives.	69

Abbreviations

AIS	Automatic Identification System
ARPA	Automatic Radar Plotting Aid
COLREGs	Convention on the International Regulations for Preventing Collisions at Sea
CPA	Closest Point of Approach
DBSCAN	Density-Based Clustering of Applications with Noise
DCPA	Distance at Closest Point of Approach
ECDIS	Electronic Chart Display and Information System
EM	Expectation Maximization
GMM	Gaussian Mixture Model
GNSS	Global Navigation Satellite System
GRU	Gated Recurrent Unit
HDBSCAN	Hierarchical Density-Based Clustering of Applications with Noise
IMO	International Maritime Organization
KL	Karhunen-Loève
LIDAR	Light Detection and Ranging
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
OOW	Officers on Watch
PCA	Principle Component Analysis
PDF	Probability Density Function
RADAR	Radio Detection and Ranging
RAE	Recurrent Autoencoder
RC	Research Contribution
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RO	Research Objective
SOLAS	Safety of Life at Sea
TCPA	Time to Closest Point of Approach
VAE	Variational Autoencoder
VHF	Very High Frequency
VRAE	Variational Recurrent Autoencoder
VTs	Vessel Traffic Service

Nomenclature

a	Activation function	\mathbf{S}_b	Between-class scatter matrix
\mathbf{A}	Transformation matrix	\mathbf{S}_m	Mixture scatter matrix
b, \mathbf{b}	Bias scaler, vector	\mathbf{S}_w	Within-class scatter matrix
BIC	Bayesian Information Criterion	\mathbf{u}	Update gate
C	Total number of classes	\mathbf{v}	Class membership vector
COG_0	Initial course over ground	\mathbf{w}, \mathbf{W}	Weight vector, matrix
d	Dimensionality	x	Relative x-coordinate [m]
D_c	Core distance	\mathbf{x}	Data
D_{KL}	Kullback-Leibler divergence	y	Relative y-coordinate [m]
D_m	Mutual reachability distance	\mathbf{y}	Output
e	Eigenvector	\mathbf{z}	Latent variable
\mathbf{E}	Eigenvector matrix	$\boldsymbol{\varepsilon}$	Gaussian noise
f	General function	$\boldsymbol{\Lambda}$	Eigenvalue matrix
g	General decoder function	$\boldsymbol{\mu}$	Mean
\mathbf{h}	Hidden representation	ϕ	Encoder parameters
\mathbf{I}	Identity matrix	π	Prior
J	Loss function	σ	Sigmoid activation function
J_s	Class separability measure	$\boldsymbol{\sigma}$	Standard deviation
K	Number of free parameters	$\boldsymbol{\Sigma}$	Covariance
L	Length	θ	Decoder parameters
LL	Log-likelihood	Θ	Model parameters
M	Total number of models		
\mathbf{n}	New candidate vector		
N	Total number of data points		
o	Perceptron output		
p	Probability density function		
q	Probabilistic encoder function		
\mathbf{r}	Reset gate		
s	Activation function variable		
\mathbf{s}_0	Initial vessel state		

Subscripts

b	Backward
c	Class
f	Forward
g	Global
h	Hidden
i	Data point
in	Initial
k	Class membership
L	Length
m	Model
n	New candidate
r	Reset
t	State
u	Update
x	Input
y	Output
z	Latent
μ	Mean
ϕ	Encoder
σ	Standard deviation
θ	Decoder

Superscripts

$\hat{}$	Estimated parameter
l	Layer
P	Total number of layers

Chapter 1

Introduction

This chapter provides an overview of the research in this thesis. First, the motivation and background of the study are presented. Next, the research objectives and scope of the thesis are introduced. The main research contributions are then outlined. Subsequently, the included publications are presented and briefly summarized. Finally, an outline of the remainder of the thesis is presented.

1.1 Motivation and Background

Modern technologies are advancing at a rapid pace, with developments in artificial intelligence, computational power and communications technologies permeating virtually every industry. Technologies e.g. image and speech recognition, that previously were inconceivable, are now commonplace on hand-held devices. Many of these developments are largely due to the success of recent advances in machine learning.

Machine learning is a sub-field of artificial intelligence, where computers are able to learn from data without being explicitly programmed. In this manner, the algorithms emulate human behavior, and their ability to learn from experience. In recent years, most of the advances in machine learning have been in a field known as deep learning (Goodfellow et al., 2016). Deep learning leverages artificial neural networks, that initially were designed to model brain functions. Machine learning techniques are data-driven, in that the model parameters are optimized by learning from the data.

Due to the ubiquity of data from various sensors, such data-driven techniques have

gained interest across a wide variety of domains, with the potential to enable safer and more efficient operations in many industries. Hermann et al. (2016) argued that the next industrial revolution, a so-called digital revolution, is taking place, known as Industry 4.0. The maritime sector is among such industries, where the shipping industry has identified the potential of utilizing recent technological developments to optimize its operations. This technological revolution in shipping is similarly being referred to as Shipping 4.0 (Rødseth et al., 2015).

In recent years, the concept of autonomy (Krogmann, 1999) has become more and more prevalent, with autonomous cars (Chan, 2017) among the most highly researched topics. These developments are in a large part facilitated by recent developments in machine learning. Similarly, machine learning is viewed as a main facilitator of autonomous ship operations. One of the primary objectives for autonomous ships is to replace the functions of the navigator. The Officers on Watch (OOW) are essential on conventional vessels, and facilitate crucial functions e.g. collision avoidance and path planning. Some aspects of navigation can also be executed under autopilot type systems. Control algorithms to facilitate autopilot functions have existed for many years, with the first automatic ship steering mechanism developed already in 1911 (Fossen, 2000).

However, the OOW rely on their degree of situation awareness (Endsley & Jones, 2012), developed through experience, to navigate in a safe and efficient manner, even when autopilots are utilized. Situation awareness can be thought of as *"Being aware of what is happening around you and understanding what that information means to you now and in the future"* (Endsley & Jones, 2012). Situation awareness is further split into three levels. Level 1 situation awareness relates to the perception of the surroundings (e.g. ship traffic). Level 2 situation awareness entails comprehension of the situation, e.g. the importance of the situation with respect to the integrity of an operation. Level 3 is the highest level of situation awareness, and relates to the projection of the future status, e.g. via simulation of future dynamics. To facilitate safe and efficient autonomous operations, technology must be developed to emulate such human situation awareness.

Enhanced Situation Awareness

It is claimed that about 75%-96% of maritime accidents can be attributed to human error (Rothblum, 2000). As a result, autonomous ship functions are argued to minimize error due to human involvement. It is further suggested that autonomous functions can serve as a decision support system to improve the safety and efficiency of maritime operations until fully autonomous vessels are derived. Autonomy in shipping has been divided into four levels, developed by the International

Maritime Organization (IMO)(IMO, 2020), where the first level aims to provide such decision support:

1. Ship with automated processes and decision support: Seafarers are on board to operate and control shipboard systems and functions. Some operations may be automated and at times be unsupervised but with seafarers on board ready to take control.
2. Remotely controlled ship with seafarers on board: The ship is controlled and operated from another location. Seafarers are available on board to take control and to operate the shipboard systems and functions.
3. Remotely controlled ship without seafarers on board: The ship is controlled and operated from another location. There are no seafarers on board.
4. Fully autonomous ship: The operating system of the ship is able to make decisions and determine actions by itself.

However, many challenges may arise when involving automation to support human decision making (Endsley, 2017). In many cases, it has been shown that automation can result in new failure modes that compound the risk associated with various operations when interacting with humans (Bainbridge, 1983; Strauch, 2017). Endsley (2017) has further argued that instead of designing automation systems to support decision making, systems should be developed to support situation awareness. As such, technology developed to facilitate situation awareness in autonomous vessels can also be used to provide enhanced situation awareness to ship navigators.

Situation awareness is a key facilitator of collision avoidance actions. The dynamic obstacles presented by other vessels likely constitute a significant challenge to most OOW, where their situation awareness must constantly be updated to maintain safe operations. This is supported by interviews with navigators in Sharma et al. (2019), where multiple aspects of ship traffic were outlined as important to facilitate situation awareness. To support navigational situation awareness, recent developments of automation functions have focused on supporting the perception of elements in the environment via existing technologies, e.g. computer vision. Endsley & Jones (2012), however, argued that the best way to support human performance is by supporting high levels of situation awareness, e.g. level 3 projections of the future states of a system.

Ship navigators likely utilize level 3 projections to simulate future ship traffic and evaluate the likelihood of close-encounter situations. In this manner, proactive collision avoidance maneuvers can be implemented to prevent such situations from

arising. Such proactive measures may for instance include minor speed or heading alterations. However, such actions must comply with relevant rules and regulations e.g. the COLREGs (Convention on the International Regulations for Preventing Collisions at Sea) (Cockcroft & Lameijer, 2011). Currently, the most common technique employed to support such predictions is a linear extrapolation of a trajectory based on the current speed and course over ground. However, in many cases, the assumption that the vessel will maintain constant speed and course over ground will not be valid, especially in regions of complex traffic e.g. inland waterways and around ports. Linear predictions are, therefore, not sufficient to support level 3 situation awareness in many cases.

Ship navigators likely conduct such simulations of the future based on their experience with historical ship behavior in the region, or based on experience with similar situations. Such experience likely facilitates higher levels of situation awareness (Endsley & Jones, 2012). Endsley & Garland (2000b) argued that expert operators rely on their ability to predict future system dynamics, and that this ability is the mark of a skilled expert. If a navigator has a high level of situation awareness, they likely possess internal models capable of conducting complex predictions of ship behavior. Technology to support autonomous vessels should, therefore, be developed to provide navigational expertise that facilitates such predictions. Such automation could, in turn, support high levels of situation awareness in ship navigators, especially in cases where the navigator is inexperienced.

Designing automation to support level 3 situation awareness, however, is not easily achieved. First, navigational experience must be artificially represented. Such experience largely relates to historical ship behavior, where it is assumed that the future behavior of a selected vessel will be similar to that of the past behavior of similar vessels. One approach to facilitate this may be to leverage historical Automatic Identification System (AIS) data. The historical AIS data for a given region outlines the historical ship behavior. As such, navigational experience can be artificially represented in historical AIS data sets. However, high levels of situation awareness require the development of models that can predict future behavior based on past behavior. To emulate this, it is suggested to leverage data-driven techniques, specifically machine learning, to facilitate enhanced maritime situation awareness.

1.2 Research Objectives

The main objective of this thesis is to enhance the safety of maritime transportation by utilizing recent developments in data driven techniques, while supporting

the further requirements for autonomous ship navigation. To facilitate this, it is suggested to identify methods to support level 3 situation awareness for ship navigators. If ship navigators are able to predict future ship traffic accurately, proactive collision avoidance measures can be taken to prevent potential close-encounter situations from arising, thereby enhancing the safety of maritime operations. It is, therefore, suggested to investigate emulating the development of human situation awareness by applying machine learning techniques to historical AIS data sets. If effective, this should yield models capable of predicting future ship trajectories. Such projections can be utilized by navigators, or future autonomous vessels, to minimize the risk of future close-encounter situations. As such, they will provide a form of enhanced maritime situation awareness. The following research objectives (RO) are formulated to support these developments:

- RO1** Leverage machine learning to provide methods to support maritime situation awareness.
- RO2** Leverage historical AIS data to provide methods to support level 3 maritime situation awareness by artificially serving as navigational experience.
- RO3** Develop methods to emulate the development of high level maritime situation awareness in humans by:
 - (i) Developing methods to categorize ship behavior.
 - (ii) Developing methods to facilitate behavior models for predicting future ship behavior.
 - (iii) Developing methods to facilitate pattern matching of observed ship behavior.

Scope of Work

The scope of the work in this thesis is constrained to methods to support level 3 maritime situation awareness. Maritime situation awareness is further constrained to relate to the situation awareness of ship navigators. The applicability of such situation awareness is also limited to collision avoidance. Furthermore, the scope is limited to two-vessel encounter situations. However, it can be expanded to a multi-vessel encounter situations. Historical AIS data will also provide the basis for all studies. As such, weather parameters will not be considered in this work. However, it is expected that the AIS data are based on ship navigator decisions that may have been influenced by prevailing weather conditions. Finally, this thesis aims solely to provide predictions to ship navigators. As such, risk evaluation metrics are not considered as part of the scope.

1.3 Research Contributions

In line with the main objective of the thesis, the research outcome provides methods to enhance the safety of maritime transportation. In focusing on level 3 situation awareness, methods are developed to support ship behavior prediction by leveraging historical AIS data in conjunction with machine learning. As such, the general contributions can be considered to be frameworks that facilitate level 3 situation awareness projections of ship behavior. However, the following research contributions (RC) are considered to be provided by the research, that together comprise such frameworks:

RC1 Methods to generate representations of historical ship behavior

RC2 Methods to cluster historical ship behavior

RC3 Methods to classify a novel ship trajectory to a cluster of historical behavior

RC4 Methods to predict future ship behavior

RC5 Methods to identify anomalous ship behavior

1.4 Appended Papers

The publications included in this thesis are listed below. The papers are presented in the following order to improve the readability of the thesis.

- (I) **Murray, B.**, Perera, L. P. (2021). Ship Behavior Prediction via Trajectory Extraction-Based Clustering for Maritime Situation Awareness. Submitted and in First Revision for Publication in *Journal of Ocean Engineering and Science*.
- (II) **Murray, B.**, Perera, L. P. (2020). A Dual Linear Autoencoder Approach for Vessel Trajectory Prediction Using Historical AIS Data. In *Ocean Engineering*, 209, 107478. <https://doi.org/10.1016/j.oceaneng.2020.107478>
- (III) **Murray, B.**, Perera, L. P. (2020). Unsupervised Trajectory Anomaly Detection for Situation Awareness in Maritime Navigation. In *Proceedings of the 39th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2020)*. ASME. <https://doi.org/10.1115/OMAE2020-18281>

- (IV) **Murray, B.**, Perera, L. P. (2021). Deep Representation Learning-Based Vessel Trajectory Clustering for Situation Awareness in Ship Navigation. Accepted for Publication in *Developments in Maritime Technology and Engineering. Proceedings of the 5th International Conference on Maritime Technology and Engineering (MARTECH 2020)*. Taylor and Francis.
- (V) **Murray, B.**, Perera, L. P. (2021). An AIS-Based Deep Learning Framework for Regional Ship Behavior Prediction. Submitted and in First Revision for Publication in *Journal of Reliability Engineering and System Safety. Special Issue on Safety of Maritime Transportation Systems*.

Papers Published by the Author but Not Included in Thesis

- **Murray, B.**, Perera, L. P. (2018). A Data-Driven Approach to Vessel Trajectory Prediction for Safe Autonomous Ship Operations. *2018 13th International Conference on Digital Information Management (ICDIM)*, 240–247. IEEE. <https://doi.org/10.1109/ICDIM.2018.8847003>
- **Murray, B.**, Perera, L. P. (2019). An AIS-Based Multiple Trajectory Prediction Approach for Collision Avoidance in Future Vessels. *Proceedings of the 38th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2019)*. <https://doi.org/10.1115/OMAE2019-95963>
- Perera, L. P., **Murray, B.** (2019). Situation Awareness of Autonomous Ship Navigation in a Mixed Environment Under Advanced Ship Predictor. *Proceedings of the 38th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2019)*. <https://doi.org/10.1115/OMAE2019-95571>

Brief Summary of Appended Papers

In **Paper I**, a method to leverage historical AIS data to cluster specific ship behavior based on locally extracted data is developed. Using these clusters of specific behavior, the future trajectory of a selected vessel can be predicted. **Paper II** builds upon the work in Paper I by introducing a dual linear autoencoder approach to facilitate trajectory predictions to support level 3 maritime situation awareness. The work in **Paper III** develops a method to facilitate anomaly detection in support of the methods in Paper I and Paper II. **Paper IV** investigates leveraging deep learning to facilitate clustering of regionally extracted historical AIS trajectories. **Paper V** builds upon this work by introducing a deep learning framework for trajectory clustering, classification and prediction.

1.5 Outline of The Thesis

In **Part I**, the methodology and context of the thesis are presented. **Chapter 2** discusses maritime situation awareness, outlining the context and motivation for the developed methods. **Chapter 3** introduces relevant machine learning techniques that can be leveraged to emulate high level human situation awareness through historical AIS data.

Part II presents the research outcome. **Chapter 4** presents a summary of the appended papers. **Chapter 5** then provides a discussion of the research, with concluding remarks and suggestions for further work in **Chapter 6**.

The included papers are appended in **Part III**.

Part I

Methodology and Context

Chapter 2

Maritime Situation Awareness

In order to facilitate safe maritime operations, ship navigators must have an adequate degree of what is known as situation awareness. Situation awareness provides the basis for risk mitigation actions e.g. collision avoidance. In this chapter, maritime situation awareness is discussed, as well as how automation can be utilized to enhance the situation awareness of ship navigators.

In the first section, the theory of situation awareness is presented. The section presents relevant literature to provide a basis for understanding the mechanisms of situation awareness. To best emulate human situation awareness, such development mechanisms must be understood. Autonomy, and how automation functions can best support human performance are also discussed. It is argued that automation should be designed to support the situation awareness of operators.

In the next section, situation awareness in ship navigation is discussed. Relevant information requirements for the three levels of situation awareness are presented, along with existing automation technology to support the respective levels. It is argued that limited support for level 3 situation awareness currently exists.

Proactive collision avoidance is then discussed in light of level 3 situation awareness projections. Applications for a long-range trajectory prediction facilitated by automation are presented, as well as how any proactive collision avoidance measures based on such predictions must adhere to existing rules and regulations. Finally, existing approaches that may facilitate such level 3 predictions are discussed.

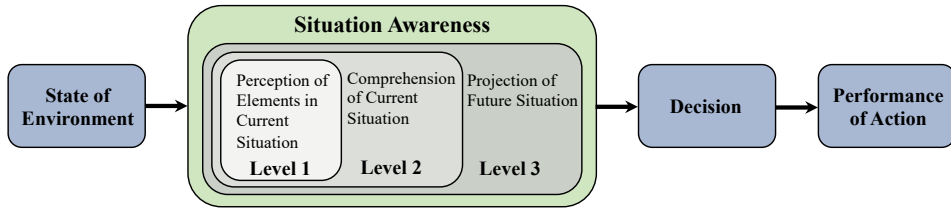


Figure 2.1: Model of situation awareness adapted from Endsley (1995).

2.1 Theory of Situation Awareness

The term situation awareness dates back to World War I, where its importance was identified in the pilot community (Endsley & Garland, 2000a). The term has since then been adopted in a wide variety of domains in which operators can enhance their performance through high levels of situation awareness. Such domains include education, driving, train dispatching and power plant operations (Endsley & Jones, 2012). A formal definition of situation awareness was outlined in Endsley (1988a) as:

"The perception of of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"

In a more general sense, situation awareness can be thought of as *"Being aware of what is happening around you and understanding what that information means to you now and in the future"* (Endsley & Jones, 2012). Such situation awareness is utilized to achieve some form of goal or objective. A model of situation awareness in dynamic decision making is illustrated in Fig. 2.1. The figure illustrates how the state of the environment provides the basis for situation awareness. This then leads to a decision, and a subsequent action in line with the objective of the operator.

Endsley (1995) decomposed situation awareness into three levels:

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

These levels increase in complexity with level 1 the most basic, and 3 the most advanced. Furthermore, the levels are iterative and feed into one another, i.e. level

3 situation awareness requires level 2 which again requires level 1, as illustrated in Fig. 2.1. Situation awareness will be discussed in light of Endsley's model in this thesis.

Once situation awareness is achieved, a decision can be made, and a subsequent action implemented. Situation awareness can be viewed as an internal model of the operator, where a representation of the environment is generated via this internal model. This internal representation can then subsequently be utilized to make a decision. Situation awareness is, therefore, separate from the decision making process, as shown in Fig. 2.1. Operators should always make the best decision possible given their skills and level of situation awareness. However, incorrect decisions occur, that may result in high risk situations and accidents e.g. ship collisions.

When investigating accidents involving human operators, incorrect decisions are often attributed to human error. Studies in various domains have shown that nearly 90% of human error is due to poor situation awareness (Endsley, 1995). As a result, it can be argued that humans do not necessarily make bad decisions in some situations, but rather misunderstand the situation. Situation awareness in this manner supports better decision making in humans.

2.1.1 Mental Modeling

It is theorized that operators actively make use of what are known as mental models to facilitate situation awareness (Sarter & Woods, 1991; Endsley, 1995). It has been argued that such mental models are key enablers of level 2 and level 3 situation awareness (Endsley & Jones, 2012). This section outlines key aspects of mental modeling, and its importance in achieving high levels of situation awareness. By investigating how human situation awareness is developed, it may be possible to emulate the development mechanisms in automation technology.

Working Memory

Once a human operator has perceived the elements in the environment (level 1 situation awareness), the information must be added to their working memory. Working memory provides the basis to process information pertaining to the current situation. Using the knowledge stored in their working memory, an operator is able to comprehend the current situation (i.e. level 2 situation awareness). Furthermore, projections of future states (i.e level 3 situation awareness), and subsequent

decisions, are made in working memory. Wickens et al. (1984) argued that predicting the future states of systems imposes a heavy load on working memory. This is argued to be due its responsibility for maintaining control over current and future states, as well as appropriate actions with respect to the future conditions. As a result, the working memory of an operator can easily be overloaded, constituting a potential bottleneck for situation awareness.

Schemata and Mental Models

Long-term memory is utilized by operators to ameliorate the challenges associated with the limited capacity of working memory. It has been argued that operators employ a component of long-term memory known as schemata to assist in achieving situation awareness (Rasmussen & Rouse, 1981; Braune & Trollip, 1982). Such schemata are frameworks that provide a basis for human understanding of information relating to complex system states and functions (Bartlett, 1932; Mayer, 1983). They can, therefore, be viewed as a compressed version of previous situations, where the most important details are encoded in a long-term memory bank. This encoding is structured in a framework to best describe the relevant element or situation. For instance, a schema of a ship would likely comprise the most important ship components. Similarly, a schema of a ship route will likely contain the most important details of the route. Such schemata are frameworks for observed situations, where the details of the framework are filled in based on current observations of the operator. In this manner, an operator does not need to retain as much information in working memory, as relevant schemata can be accessed and utilized to comprehend the situation.

Schemata are closely related to the concept of mental models. Mental models were defined in Rouse & Morris (1985) as "*Mechanisms whereby humans are able to generate descriptions of system purpose and form, explanations of system functioning and observed system states, and predictions of future states*". It was argued that experts develop mental models that are able to generate abstract codes from the true representations. Mental models can, therefore, be thought of as complex schemata utilized by operators to model system behavior (Endsley, 1995).

Pattern Matching

Pattern matching is an important aspect of mental models. When exposed to a situation, an operator will attempt to match stored schemata in their long-term memory to the observed situation. Schemata in this sense represent prototypical situations,

where operators recognize similar characteristics in the current situation. Endsley (1995) argued that the key to using mental models to achieve situation awareness lies in the ability to identify key features in the environment that map to key features in the schema, i.e. pattern matching. This allows for high level situation awareness without loading working memory.

Humans are highly skilled in pattern matching, and can classify a situation to a schema virtually instantaneously. Once the relevant schema is identified, an operator can use the mental model to direct their attention to key elements in the environment, as well as comprehend the situation and predict the future future states. Development of such models, however, requires experience. It is self evident that novice operators will not perform as well as expert, ie. experienced, operators. As operators are exposed to recurrent situations, they will discover recurrent causal relationships and components. Endsley (1995) argued that this forms the basis for early schema and model development in novice operators.

Model Development

A detailed outline of the development of mental models was described in Holland et al. (1986). An overview of model development is illustrated in Fig. 2.2. The authors argued that the first step in model development is learning to categorize input. In this manner, individuals learn categorization functions that map real world inputs to a representation category. Such categories can be thought of as being analogous to schemata. As individuals become more experienced, they begin recognize common characteristics between various situations or objects, i.e. patterns, and categorize them. Future situations or objects can then be classified to one of these categories via pattern matching.

The second step of model development is argued in Holland et al. (1986) to be the development of transition functions that model how objects or situations vary over time. Each category will, therefore, have its own behavior model. Such mental models are refined by comparing the predictions of the models to real life observations. With experience, individuals progressively refine model predictions against observations. It is argued that in this process, a higher number of categories are generated. This increase can be attributed to larger categories with general behavior being split into smaller categories with more specific behavior. Such specific categorizations allow the operator to develop more detailed models relating to the specific behavior of that category, enhancing their ability to predict future states with a high degree of precision.

The development of mental models can, therefore, be argued to be viewed as three-

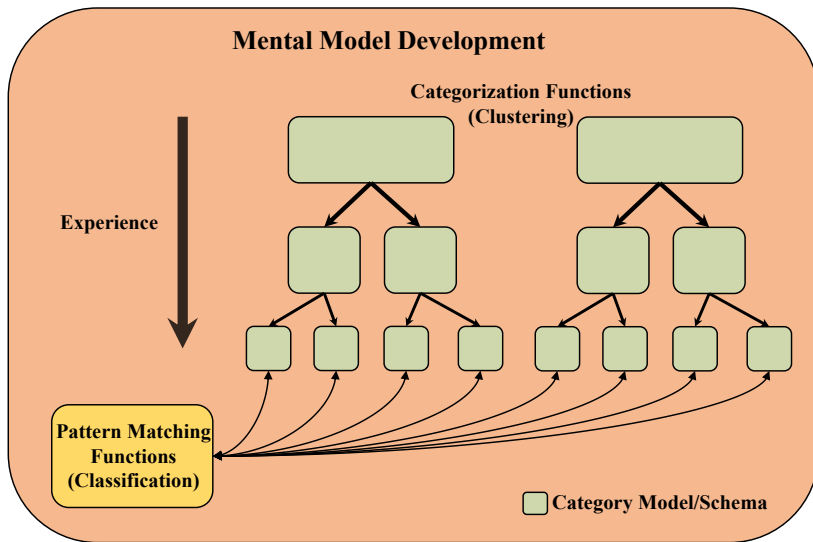


Figure 2.2: Development of mental models.

fold. First, individuals develop categories of experiences. This is analogous to a clustering of past situations, where similar situations will be grouped together in the same cluster. Second, classification functions are developed to classify novel situations to one of the existing clusters. Third, models are developed to describe the behavior in each cluster. These facilitate level 2 and level 3 situation awareness (comprehension and prediction). When applying such mental models, an operator will observe a novel situation and classify it to one of the existing categories. Using the relevant behavior model, they are able to comprehend the novel situation as well as predict future behavior without loading working memory.

Situation Models

It is clear that mental models are powerful tools that facilitate situation awareness. Figure 2.3 illustrates the relationship between mental models and situation models. A situation model can be thought of as the current state of the mental model (Endsley & Garland, 2000b), i.e. the schema is filled in with relevant details pertaining to the current situation. Pattern matching is utilized to select the appropriate schema, and associated behavior model, that matches the situation. The mental model will then direct the attention of the operator to critical aspects of the environment. Such models also aid in integrating relevant elements to facilitate comprehension of the situation (level 2 situation awareness). Furthermore, the selected behavior model

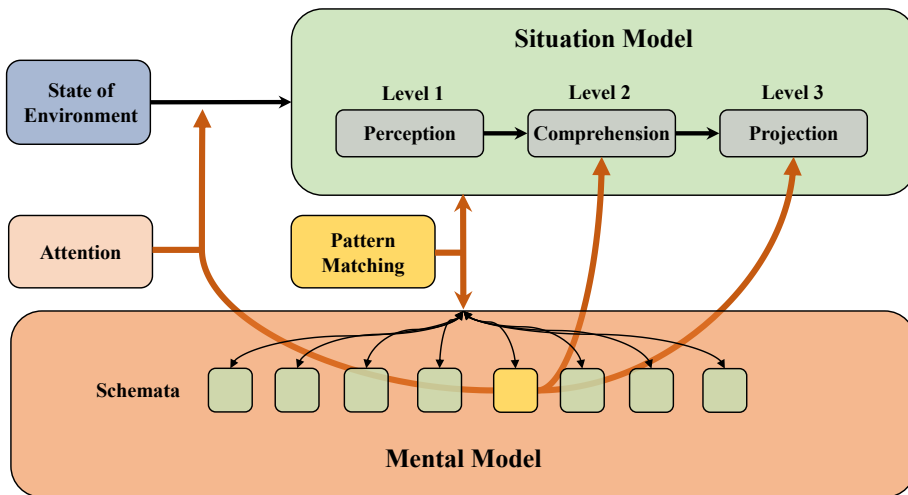


Figure 2.3: Role of mental models adapted from Endsley & Garland (2000b).

will model the dynamic behavior of the situation. In this manner, the model can be used to predict future dynamics, i.e. projection (level 3 situation awareness). Via computational algorithms, such models towards predicting situation dynamics may be replicated, facilitating high level situation awareness to autonomous systems.

2.1.2 Autonomy

Autonomous capabilities are being developed at an increasing rate with the purpose of reducing the workload of human operators as well as increasing the level of safety associated with the systems in many domains. These include the maritime, automotive and aviation industries (Endsley, 2017). Most relevant research related to autonomy has historically focused on automation. Recently, however, the term autonomy has become more prevalent, and the terms are often conflated.

Endsley (2017) discussed the work in Krogmann (1999), that argued that autonomy differs from automation in that autonomous systems are designed to function independently for large periods of time without the ability for external intervention. Furthermore, autonomous systems were argued to leverage intelligent algorithms

that are capable of learning and adapting to unforeseen and dynamic situations. Automation, however, was argued to rely on logic-based programming, making it less flexible in the face of uncertainty. In this sense, autonomous systems can deal with situations that were not explicitly pre-programmed, rendering them more intelligent. Autonomy can, therefore, be viewed as an evolution of automation that historically has been more limited with respect to its capabilities (Hancock, 2017).

Autonomy is driving technology development in many cases, e.g. technology to support autonomous ships. It is assumed, however, that fully autonomous ship operations will not be common for some time. Semi-autonomous operations involving human operators are argued to be the most prominent form of autonomy in many domains (Endsley, 2017). Such semi-autonomous operations may for instance include remote-controlled operations, or operations where the ship is navigating autonomously, but being monitored closely by human operators. Endsley & Jones (2012) developed a taxonomy of various levels of automation, ranging from manual control to full automation.

Decision Support

One of the levels of automation in the taxonomy of Endsley & Jones (2012) related to systems that could provide decision support functions. This is a common argument made for the development of autonomous technology, as it can be argued to additionally function as a decision support system to human operators. According to the outlined taxonomy, decision support implies a computer generating recommended options for the human to choose between, whilst also allowing the human to override the system and input their own choice. The premise of such a system is that intelligent automation functions can improve human decision making by advising operators on what to do in various situations, especially in cases where the operator has limited experience (Endsley & Jones, 2012).

However, utilizing automation for decision support has been found to be problematic in many studies as discussed in Endsley (2017). One issue related to automation-based decision support was argued to be a decision-biasing effect (Croll & Coury, 1990; Sarter & Schroeder, 2001; Lorenz et al., 2002; Reichenbach et al., 2011; Endsley & Jones, 2012). In this case, the automation may recommend a course of action that significantly biases the decision of the operator. In cases in which the automation is correct, such systems are shown to assist in correct decision making. However, in cases where the decision is incorrect, human operators have been found to perform worse than they would have without any support (Layton et al., 1994; Olson & Sarter, 1999). This effect has been found to be intensified in cases where the system is considered to be very reliable (Metzger &

Parasuraman, 2005; Rovira et al., 2007). Furthermore, it has been found that decisions are slowed when using decision support systems as operators require extra time to compare decisions output by the system to their own understanding of the situation (Endsley & Kris, 1994; Madhavan & Wiegmann, 2005). Many of these studies have focused on aircraft pilots. It can be argued that ship navigators and aircraft pilots share many similarities, and as such the research is transferable to some extent.

Situation Awareness Support

As opposed to providing direct decision support, it has been argued that automation can be beneficial in improving human performance through supporting situation awareness (Endsley & Jones, 2012). Situation awareness support is presented as a lower level of automation than decision support in the taxonomy from Endsley & Jones (2012). With respect to level 1 situation awareness, automation can provide methods to collect and present relevant information to the user. Furthermore, systems can be designed to integrate information to support comprehension and projection (level 2 and level 3). In this manner, the operator is still highly involved in the decision making process, reducing the aforementioned issues related to automation designed to provide decision support. Endsley (2017) argued further that automation should be designed to support situation awareness as studies have shown that such systems can significantly reduce the workload of the user, as well as enhance situation awareness and performance with little negative effects (Endsley, 1988b; Sarter & Schroeder, 2001; Onnasch et al., 2014; Endsley & Jones, 2012).

2.2 Situation Awareness in Ship Navigation

As discussed in Sec. 2.1.2, automation functions should be developed to support situation awareness. Situation awareness is predicated upon the relevant goal or objective of the operator. The primary objective of a ship navigator is to navigate the vessel to its destination in a safe and efficient manner. The integrity of the operation must be maintained at all times, as well as adherence to relevant rules and regulations. It can be argued that one of the main challenges facing navigators in achieving this goal is effective collision avoidance. Collision avoidance is defined in Huang et al. (2020) as a process in which one ship departs from its planned trajectory to avoid a potential undesired physical contact with another ship at given point in the future. In this context, the ship under control is defined as the own ship.

The dynamic obstacles represented by other ships are known as target ships.

Sharma et al. (2019) investigated the situation awareness information requirements of ship navigators. The results of the study indicated that information relating to ship traffic was essential in achieving all three levels of situation awareness. The findings support the argument that situation awareness in ship navigation supports collision avoidance decisions and actions. Other challenges, e.g. grounding, do exist, but the dynamic obstacles inherent in target ships navigating in close proximity to the own ship can be argued to constitute significant challenges for navigators with respect to situation awareness requirements. As a result, maritime situation awareness will be discussed in the context of collision avoidance for the case of ship navigation.

Enhancing the situation awareness of the OOW to facilitate effective collision avoidance is a classic research topic (Huang et al., 2020), with a myriad of technologies developed to support navigators. Recently, research into autonomous systems to replace human functions has gained much attention. Autonomous ships will need to achieve their own level of situation awareness to conduct effective collision avoidance maneuvers. As such, technology should be developed that can emulate human behavior (Perera, 2020). It is additionally argued that the technology developed to facilitate situation awareness for autonomous ships can benefit manned vessels (Huang et al., 2020). Such automation technologies should enhance the situation awareness of navigators, compared to the degree of situation awareness achievable without such tools. In this section, navigational situation awareness requirements are discussed, as well as relevant automation to support maritime situation awareness.

2.2.1 Level 1 Situation Awareness

Information Requirements

The case of ship navigation fits well into the architecture illustrated in Fig. 2.1. The state of the environment in this domain relates to the current environmental conditions e.g. wave height, tide, wind speed, current speed and their relative directions. Furthermore, visibility, under-keel clearance, local geography, and fairway geometry constitute crucial environmental elements for a navigator's situation awareness. Such environmental conditions will, for instance, influence the maneuverability of the vessel, and need to be taken into consideration by the navigator. These environmental conditions can be viewed as quasi-static, as they are dynamic with time, but are near constant with respect to the horizon of a navigator's decision making,

and should, therefore, generally be easily predicted in time.

Dynamic obstacles, however, also exist in ship navigation. These primarily relate to the surrounding maritime traffic. The current position of other vessels in the region, as well as their course over ground and speed over ground, provide critical information to the navigator to facilitate situation awareness. Sharma et al. (2019) identified information pertaining to ship traffic and obstacles as necessary elements to achieve level 1 situation awareness, where the location and number of targets were outlined as important. Such dynamic obstacles are likely to pose a significant challenge to a navigator, as they must be capable of implementing effective collision avoidance actions to maintain the integrity of the operation. Adequate perception of such obstacles is, therefore, important in ship navigation. The perception of all such environmental conditions constitutes level 1 situation awareness in Fig 2.1.

Automation Support

Much of the technological development towards aiding maritime situation awareness has focused on supporting level 1 situation awareness for navigators. In order to perceive the relevant targets, navigators rely heavily on visual observation, in addition to the navigational tools available to them. Such tools include radar facilitated by ARPA (Automatic Radar Plotting Aid), conning display, AIS and ECDIS (Electronic Chart Display and Information System). Perera & Guedes Soares (2015) argued that the best navigational tools should be available to navigators to support them in identifying relevant obstacles.

There has recently been a significant amount of research conducted on technology to facilitate level 1 situation awareness to autonomous vessels through electro-optical sensors e.g. stereo cameras, RADAR (Radio Detection and Ranging) and LIDAR (Light Detection and Ranging) (Yang et al., 2017; Prasad et al., 2017; Bloisi et al., 2017; Cane & Ferryman, 2018). Computer vision techniques that leverage machine learning have been shown to be able to detect and classify various obstacles, thereby facilitating level 1 situation awareness. Within the automotive industry, these techniques are viewed as an enabler for autonomous cars. Techniques e.g. semantic segmentation (Trembl et al., 2016), that classify pixels as belonging to various classes (e.g. road, side walk, pedestrian, car etc.), facilitate an awareness with respect to the surroundings of the autonomous car. Many of the techniques developed towards autonomous cars are also integrated into support systems for drivers e.g. obstacle detection. Similarly, technology towards autonomous shipping can be utilized to support navigators via information presentation.

2.2.2 Level 2 Situation Awareness

Information Requirements

Level 2 situation awareness requires that all relevant elements in the current situation have been perceived, i.e. sufficient level 1 situation awareness. Based on this awareness, a navigator can comprehend the current situation, and the implications it has for the safety of the vessel. Sharma et al. (2019) found that the deviation between the ideal and current system states, as well as impact of events on navigation were necessary to achieve level 2 situation awareness. Among the relevant information required for level 2 situation awareness, the study identified the current separation between the own ship and target ships, as well as the distance to the nearest obstacles as relevant information requirements. Furthermore, the impact of traffic conditions, ship maneuvers, alteration of course and speed were identified. With respect to collision avoidance, it can be argued that the current risk of collision is evaluated at this level. Parameters relating to the Closest Point of Approach (CPA), e.g. the Time to Closest Point of Approach (TCPA) and Distance at Closest Point of Approach (DCPA), have generally been utilized to evaluate the collision risk (Huang et al., 2018). The navigator should, therefore, have enough knowledge and experience to evaluate the current collision risk given their perception of the situation under the respective parameters within a reasonable time frame.

Automation Support

Automation support with respect to level 2 situation awareness involves evaluating the current collision risk. The ARPA facilitates DCPA and TCPA calculations, and can be considered to be the most common tool to support level 2 situation awareness with respect to collision avoidance. Furthermore, AIS information integrated into the ECDIS provides the navigator with an overview of the traffic congestion as well as the region available for maneuvering. Such evaluations can also be thought of as being the domain of level 3 situation awareness (projection). The transition with respect to collision avoidance is, however, slightly fuzzy in this case. The current situation can be argued to be comprised of the current collision situation given the relative speed and course over ground of the own ship and target ships. As such, level 2 situation awareness is assumed to not predict future maneuvers. Such predictions are considered to be the domain of level 3 situation awareness in this study.

2.2.3 Level 3 Situation Awareness

Information Requirements

The highest level of situation awareness in Endsley's model is level 3. Navigators that are able to achieve level 3 situation awareness are able to forecast future events and dynamics. This ability likely allows navigators to make timely decisions to minimize the risk of collision. Endsley & Garland (2000b) argued that experienced operators rely heavily on their ability to predict the future. This argument is likely valid for the case of the maritime domain as well, where experienced ship navigators are more likely able to predict future ship behavior accurately, and can leverage this ability to conduct proactive measures to reduce collision risk.

In a collision avoidance setting, level 3 situation awareness primarily entails predicting the future trajectory of target ships, in addition to the future trajectory of the own ship. Based on these predictions, the future risk of collision can be evaluated. This is supported by the results found in Sharma et al. (2019), in which navigators identified the projected position of the own ship, the projected movement of target ships, and traffic density as relevant information to achieve level 3 situation awareness.

Automation Support

Currently, there is limited automation support for level 3 situation awareness with respect to collision avoidance actions. Automation on board vessels that calculate the DCPA and TCPA utilize linear predictions of the future trajectory of target vessels based on their current speed and course. The speed and course through water are calculated via the ARPA, whereas the the values over ground are reported via AIS. In cases where the TCPA is low enough, such linear predictions will be meaningful, as the collision risk will be high if no alterations are made. Additionally, in open waters, such predictions may be accurate. However, with respect to inland waterways and near ports, the ship traffic will likely be far more complex and congested.

In an excerpt from an interview conducted in Sharma et al. (2019), it was stated that level 3 situation awareness entailed *"If there is any traffic nearby. If somebody's going to come, or if I'm going to meet someone at some point"*. This indicates a more long-term trajectory prediction, that is not easily facilitated by a linear prediction in complex waterways. Navigators likely rely on their experience and mental models to conduct such predictions. With more experience, especially with

respect to local traffic patterns, it is likely that navigators will be able to conduct such predictions more accurately, allowing them to take proactive measures to minimize the risk of future encounter situations. Advanced automation systems to conduct accurate predictions are, however, not available yet.

2.3 Proactive Collision Avoidance

Collision avoidance is a thoroughly studied topic, with a variety of methods suggested to evaluate collision risk, and facilitate effective collision avoidance (Tam et al., 2009; Goerlandt et al., 2015; Huang et al., 2020). The IMO outlined rules and regulations regarding collision avoidance in the COLREGs. These rules are of general applicability to all ocean-going vessels, where local regulations may come in addition.

Endsley & Jones (2012) argued that the ability to project the possible future states of a system (level 3 situation awareness) was a critical skill in many domains. For instance, it allows operators to create contingency plans, and supports proactive decision making. Proactive collision avoidance actions are likely based on level 3 projections as discussed in Sec. 2.2.3. If effective, such actions have the potential to increase the safety associated with ship navigation, as navigators can be made aware of potential collision situations far in advance, and take early action to prevent them from arising. However, any such actions must comply with the COLREGs. A thorough guide to the COLREGs can be found in Cockcroft & Lameijer (2011). In this section, the process of collision risk assessment and avoidance are discussed in light of the COLREGs and level 3 situation awareness-based proactive measures.

2.3.1 Vessel Encounter Situation

Collision risk is addressed in Rule 7 of the COLREGs, where it is outlined that when two vessels in sight of each-other approach one another with no apparent alteration of compass bearing, this entails a risk of collision. If collision risk is deemed to exist between two vessels, they are considered to be in an encounter situation. In such a case, one vessel will be designated the give-way vessel and one the stand-on vessel.

Collision Risk Assessment

In practice, the most common collision risk evaluation techniques are based on the CPA. The CPA is an estimate of the closest position of an approaching ship. Two indicators are primarily used with respect to the CPA, the distance (DCPA) and time (TCPA). When these indicators are smaller than predetermined thresholds, risk of collision is deemed to exist (Huang et al., 2018). Fujii & Tanaka (1971) and Goodwin (1975) also introduced the concept of the ship domain, where an infringement of a safety region surrounding the own ship or target ship indicates a risk of collision.

Figure 2.4 illustrates a flow chart of collision risk evaluation from Tam & Bucknall (2010). Initially, the planned navigation path of the own ship is discretized at a regular interval. The future trajectory of the target ship is then predicted using a linear extrapolation of the initial velocity vector (i.e. linear prediction using constant speed and course over ground). For each time point, the CPA is evaluated. If there is an encounter situation, a safety zone relating to the ship domain may be evaluated. If the own ship position infringes upon the safety zone of the target vessel, a risk of collision is deemed to exist. If not, no risk of collision is deemed to exist. This process repeats for all time steps in the prediction. Furthermore, if vessels are moving in parabolic type trajectories, the collision risk predictions may be further complicated.

Give-way Vessel

Rule 16 of the COLREGs outlines the action by a give-way vessel. The give-way vessel is that which is directed to keep out the way of another vessel as far as possible. The vessel designated as the give-way vessel will vary based on the situation. Early and substantial action should be taken by the vessel to keep well clear. Rule 8 of the COLREGs outlines the action to avoid collision. It states that any alteration of course or speed should be large enough to be readily apparent to any other vessel. As a result, minor alterations of heading or speed are not permitted once collision risk is deemed to exist.

Stand-on Vessel

Rule 17 of the COLREGs addresses the action by the stand-on vessel. It is stated that when one of two vessels is required to keep out of the way (give-way vessel), the other shall maintain their course and speed (stand-on vessel). As such, no ac-

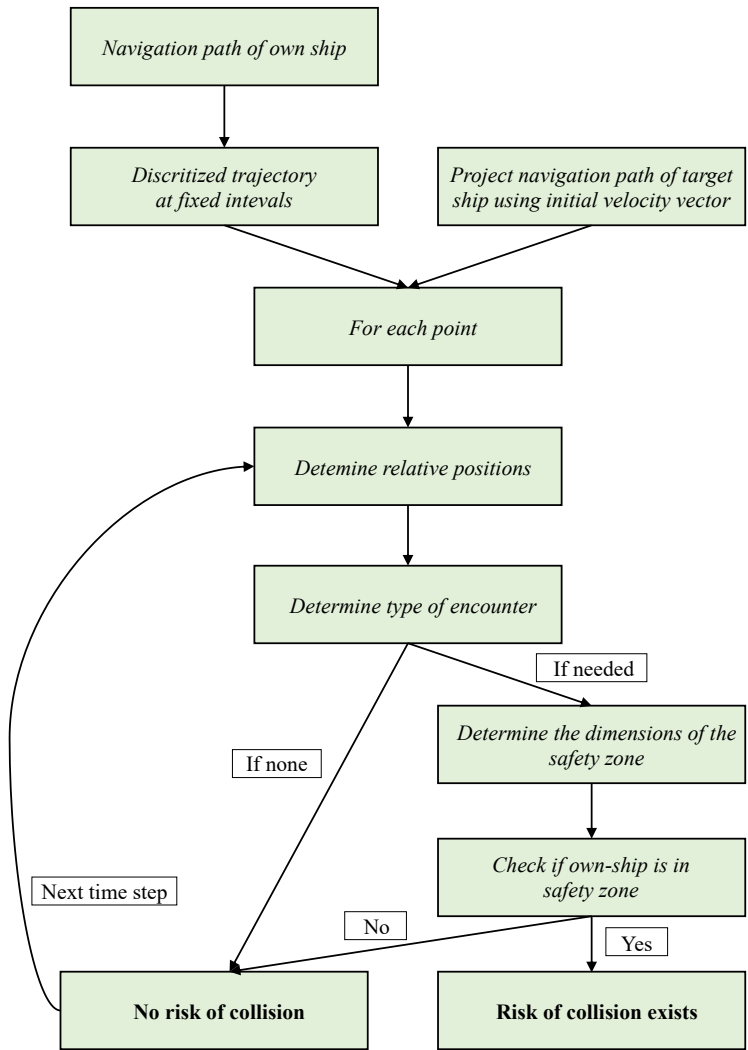


Figure 2.4: Flow chart of collision risk evaluation adapted from Tam & Bucknall (2010). Collision risk is evaluated from the perspective of the own ship with respect to a target ship.

tions are permitted by the stand-on vessel once risk of collision is deemed to exist. The stand-on vessel may, however, take action to avoid collision when it becomes apparent that the give-way vessel is not taking proper action in compliance with the rules. Rule 17 also states that when the stand-on vessel finds herself so close to collision that it cannot be avoided by the give-way vessel’s action alone, the stand-on vessel is required to take such action as to avoid collision.

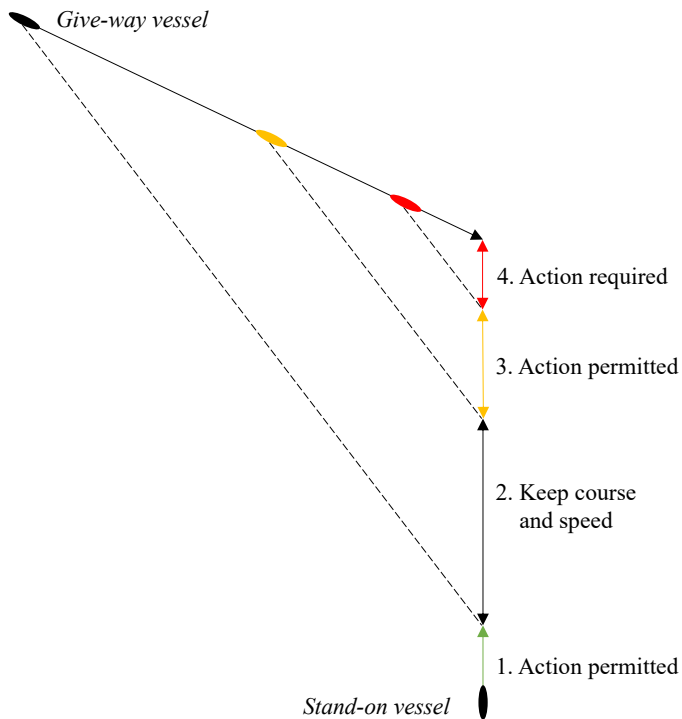


Figure 2.5: Collision situation stages adapted from Cockcroft & Lameijer (2011). A crossing situation in open seas is used for illustration, with permissible actions by the stand-on vessel.

Collision Situation Stages

A general collision situation was summarized in four stages in Cockcroft & Lameijer (2011), illustrated in Fig. 2.5.

1. At long range, before risk of collision is deemed to exist, both vessels are free to take any action.
2. Once risk of collision begins to apply, the give-way vessel must take early and substantial action to pass at a safe distance. The stand-on vessel must maintain their heading and speed.
3. If the give-way vessel does not take timely and substantial action, the stand-on vessel is permitted to avoid collision by their maneuver alone, but is required to signal their intentions to do so. Such a maneuver should not alter their course to port (as per the COLREGs).
4. When a collision cannot be avoided by the give-way vessel alone, the stand-on vessel is required to take any such action that will best avoid collision.

2.3.2 Long-Range Trajectory Prediction

Level 3 situation awareness projections are likely conducted by navigators to predict if encounter situations will arise, as argued in Sec. 2.2.3. Given a mental model that is able to accurately conduct a long-range prediction of the future trajectory of a target ship, the OOW can determine if risk of collision may exist at some point in the future. Proactive collision avoidance measures can be enacted to minimize the risk of an encounter situation from arising. Such actions may include minor speed or heading alterations that prevent the future trajectories of both vessels from intersecting. However, in a collision situation, such actions must be substantial if the vessel is the give-way vessel (Rule 8), such that it is clear to the stand-on vessel that collision avoidance actions are being taken. Minor alterations of course and speed are, therefore, only permitted in stage 1 in Fig. 2.5. Furthermore, the stand-on vessel is required to maintain their course and speed once risk of collision is deemed to exist. Any proactive collision avoidance measures that make use of level 3 situation awareness should, therefore, be enacted in stage 1 for the stand-on vessel as well. In this subsection, examples of applications of such long-range predictions based on level 3 situation awareness are presented. Finally, automation support that facilitates such long-range predictions is suggested.

Applications

Traffic Congestion One application of long-range predictions of ship behavior is to estimate future traffic congestion. By avoiding situations with a large number of vessels in close proximity, the OOW will minimize the risk of close-range encounter situations. Furthermore, the room to maneuver will likely be reduced in such situations, further compounding the level of risk. Based on a projection of the future traffic, the OOW can make proactive decisions with respect to when they will arrive at various points along the planned route. In this manner, situations with high congestion can be avoided.

Crossing Situation Crossing situations are high risk situations with respect to the potential for collision. A crossing situation is illustrated in Fig. 2.5. If such a close-encounter situation can be predicted far in advance, the own ship can take proactive measures to avoid the situation. By conducting minor speed or course over ground alterations for instance, the OOW can prevent a crossing situation from arising. Such actions must be conducted in stage 1 in Fig. 2.5.

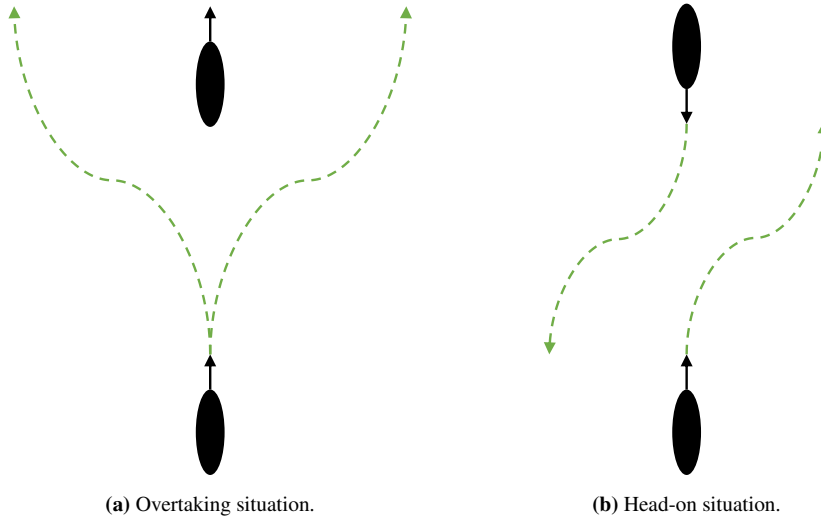


Figure 2.6: Examples of encounter situations.

Rule 15 of the COLREGs governs crossing situations, where it is stated that in such a situation, the vessel which has the other on their starboard side must keep out of the way of the other. However, in bending fairways, vessels may come into a crossing situation. In this case, each vessel must comply with Rule 9d of the COLREGs, which requires vessels to keep as near as possible to the outer limit which lies on her starboard side (Cockcroft & Lameijer, 2011). In such a case, proactive collision avoidance actions will not be necessary.

Overtaking Situation Rule 13 of the COLREGs governs overtaking situations. It is stated that any vessel overtaking another shall keep out of the way of the vessel being overtaken. Fig. 2.6a illustrates an overtaking situation. Using a long-range prediction, the OOW can plan the most optimal point to overtake another vessel. Such a maneuver should be conducted such that a safe distance can be maintained. As such, regions with constricted maneuverability should be avoided.

Head-on Situation In Rule 14 of the COLREGs it is stated that when two vessels meet on reciprocal, or nearly reciprocal courses, such that risk of collision exists, they are deemed to be in a head-on situation. This is illustrated in Fig. 2.6b In such a case, Rule 14 requires that both vessels alter their course to starboard such as to pass each other on the port side. The OOW can utilize a long-term projection of ship traffic to determine the risk of meeting other vessels in a head-on situation, and plan such that the risk of such situations is minimized. For instance, the OOW

can plan to pass another vessel in a region with adequate maneuverability such as to minimize the risk of a head-on situation.

Automation Support

Endsley & Jones (2012) argued that the best way to support human performance is to support a high level of situation awareness, e.g. the ability to project the possible future states of a system (level 3 situation awareness). However, it has been argued that some operators are just not good at conducting mental simulations of the future. This may be due to poor mental models, lack of attention, or memory limitations (Endsley, 1995). To support decision making, Endsley & Jones (2012) argued that systems should support human-system symbiosis. One method was outlined as developing systems that supported situation awareness through calculations of level 3 projections. As such, projections of the future are calculated by an automation system, and presented to the user. To support level 3 situation awareness in ship navigation, it is suggested to develop automation tools to project the future behavior of other ships. If an automation system could be developed to facilitate predictions supporting level 3 situation awareness, proactive collision avoidance actions could be taken to prevent encounter situations from arising, e.g. minor speed or heading alterations.

Whether the own ship is the give-way or stand-on vessel, proactive collision avoidance actions must take place during stage 1 of a collision situation as per Fig. 2.5. The distances at which the various stages will apply however, will vary considerably (Cockcroft & Lameijer, 2011). This may be due to the size, speed and maneuverability of the vessels, as well as local environmental conditions e.g. fairway geography and metocean conditions. According to Cockcroft & Lameijer (2011), it is suggested that the outer limit of stage 2 (i.e. when risk is deemed to exist) is 5-8 nautical miles. This condition is for the open sea, where ships are assumed to have near linear trajectories (i.e. constant course and speed). Assuming a conservative average speed of 15 kn for ships in the open sea, this corresponds to a TCPA of approximately 20-32 minutes. In more complex waterways, this value will likely be lower. It is also unlikely that navigators conduct mental simulations of ship traffic more than 30 minutes into the future when evaluating proactive collision avoidance measures, due to the uncertainty of future ship behavior. Therefore, it is suggested that automation should support ship behavior predictions up to 30 minutes into the future.

2.4 Ship Behavior Prediction

Perera & Murray (2019) introduced the concept of an advanced ship predictor to support situation awareness in future autonomous vessels. In this architecture, two levels of ship behavior predictors were suggested; a local scale predictor and a global scale predictor. It was argued that local scale predictions should support collision avoidance in close-range encounter situations. Techniques to facilitate such predictions e.g. Perera et al. (2012); Perera (2017b); Eriksen et al. (2018) can improve the accuracy of ship behavior predictions during collision avoidance maneuvers at close range, enhancing the situation awareness of the navigator. Such methods, however accurate for short periods, cannot predict trajectory changes due to future maneuvers (Lefèvre et al., 2014).

Long-range predictions to support level 3 situation awareness must take into account potential vessel maneuvers. One method to facilitate such predictions is to utilize historical traffic patterns. Given the assumption that the future behavior of a selected vessel will be similar to that of past behavior in the same region, one can infer the future ship trajectory. Historical AIS data provide insight into historical ship behavior that can be leveraged for such predictions. Various studies have utilized AIS data for vessel trajectory predictions. Ristic et al. (2008); Mazzarella et al. (2015) utilized a particle filters to facilitate trajectory predictions, and Pallotta et al. (2014); Millefiori et al. (2016); Üney et al. (2019) investigated using the Ornstein-Uhlenbeck stochastic process. These techniques, however, focus on predictions in the order of hours. As a result, they support general maritime traffic forecasting more than level 3 situation awareness-based proactive collision avoidance actions. Deep learning techniques, e.g. Forti et al. (2020), have also been investigated, where the methods were should to outperform Ornstein-Uhlenbeck approaches. Other methods e.g. Hexeberg et al. (2017); Dalsnes et al. (2018) utilized nearest-neighbor techniques, focused on aiding collision avoidance actions for autonomous ships. Rong et al. (2019) also developed a probabilistic approach on a scale relevant for collision avoidance using a Gaussian process model.

Some approaches, e.g. Pallotta et al. (2013, 2014); Mazzarella et al. (2015), apply a threefold technique to facilitate predictions. First, historical AIS trajectories are clustered into route patterns. Second, a novel trajectory is classified to one of these routes. Finally, a prediction is conducted along the route. Parallels can be drawn to the development of mental models discussed in Sec. 2.1.1. With experience, operators develop categorization functions that group similar situations together (i.e. trajectory clustering). Via pattern matching, a novel situation is classified to one of these groups (i.e. classification). Using the model pertaining to that category, future dynamics can be simulated (i.e. prediction). The experience of a navigator

relates to previously observed ship behavior for a given region. Artificially, this experience can be represented by historical AIS data.

Chapter 3

Machine Learning

In this chapter, machine learning techniques to facilitate categorization, pattern matching and prediction using historical AIS data are presented. In the first section, machine learning for enhanced maritime situation awareness is introduced, as well as the general approach facilitated by the contributions in the appended papers. The next section presents the historical AIS data set. The subsequent section outlines relevant dimensionality reduction techniques that consist of eigendecomposition-based approaches. These facilitate meaningful representations of the historical data. Next, methods utilized to cluster representations of historical ship trajectories are introduced. Anomaly detection is then briefly discussed in the following section. Finally, deep learning-based techniques are presented to generate meaningful representations of the historical data, as well as predict future behavior.

3.1 Machine Learning for Enhanced Maritime Situation Awareness

This thesis builds upon the previous work in the field leveraging historical AIS data to facilitate trajectory predictions. However, the developed methods in this study aim specifically to facilitate level 3 situation awareness projections that may aid in proactive collision avoidance. To achieve this, it is suggested to emulate high level situation awareness in humans through machine learning.

Nature has been the inspiration for developments in many fields e.g. robotics and artificial intelligence. This thesis aims to leverage machine learning, a sub-field of artificial intelligence, to support level 3 maritime situation awareness. Machine

learning allows models to learn from data without being explicitly programmed. Further, it is found that machine learning techniques mirror the manner in which it is theorized that humans develop situation awareness.

In general, machine learning is separated into supervised and unsupervised learning. Unsupervised learning relates to techniques where labels are not available for the data. In such a case, it is of interest to discover the underlying structure in the data, e.g. natural groupings. Such groups can be thought to represent various categories, or classes, of data. Discovering such groupings is referred to as clustering, where each category is a cluster. Parallels can be drawn to how humans develop mental models that employ categorization functions. As a result, clustering techniques are suggested to emulate such aspects of human situation awareness.

In supervised learning, the labels for the data are available. Such techniques can again be considered to be applied in either classification or regression applications. Classification entails pattern matching to a given category or class. If unavailable, such class labels can be estimated via unsupervised learning techniques. As outlined in 2.1.1, humans are theorized to utilize such pattern matching between observed situations and a number of developed categories or schemata. In this manner, machine learning techniques can be utilized to facilitate these functions. Regression applications can also be viewed as synonymous to the behavior models developed by humans. These models are used to predict future dynamics pertaining to a given category. In this case, the labels relate to the output of the regression model.

Such clustering, classification and prediction functions are developed through experience in human ship navigators. To artificially represent this experience, it is suggested to investigate applying machine learning methods to historical ship behavior data, available in historical AIS databases. However, such historical trajectories are high dimensional, and may degrade the performance of various algorithms. In many cases, it is useful to generate representations of the data, and thereby reduce the dimensionality. As discussed in 2.1.1, it is theorized that humans also generate such internal representations of situations that are then further processed.

General Approach for Trajectory Prediction

To facilitate level 3 situation awareness, the developed methods aim to support long-range trajectory predictions for a selected vessel up to 30 minutes into the future. In general, the following basic steps are employed to facilitate such predictions:

1. Extract relevant historical AIS trajectory segments.
2. Generate representations of the trajectory segments.
3. Cluster the trajectory representations.
4. Classify a selected vessel to one (or multiple) cluster(s).
5. Predict the future 30 minute trajectory with respect to the ship behavior in the relevant cluster(s).

By clustering the data into groupings of historical ship behavior, models can be created for the specific behavior in each cluster in a similar manner to the development of mental models. In many cases, the extracted historical ship behavior may diverge significantly. As such, the performance of a prediction model trained on all underlying data may be degraded. Trajectory predictions aimed to support level 3 situation awareness for proactive collision avoidance should, however, be as accurate as possible. Hence, by employing models of specific behavior, predictions of enhanced fidelity should be facilitated.

This is supported by the arguments made in Sec. 2.1.1, where it was suggested that operators are able to generate a higher number of categories (see Fig. 2.2) with more experience. The behavior models related to these specific categories should yield more accurate predictions of the future than categories of more general behavior.

The following sections describe the tools employed to facilitate these functions via machine learning and historical AIS data. The methods are validated by experimentation through simulation. By extracting training and test sets from the historical data, the models can be trained on the training data, and validated against the test set.

3.2 Historical AIS Data

The IMO, through the Safety of Life at Sea (SOLAS) convention, requires all vessels with a displacement over 500 tonnes in domestic waters, vessels over 300 tonnes in international waters, as well as all passenger vessels, to carry AIS transponders (Lee et al., 2019). The AIS was introduced to compliment RADAR-based observations by transmitting ship positions determined locally, e.g. via the Global Navigation Satellite System (GNSS). In this manner, the positions of ships that are not in sight, or subject to RADAR shadow, can be determined. Such information supports situation awareness and collision avoidance actions.

Table 3.1: AIS data.

Static	Dynamic	Voyage Related
MMSI	Navigational status	Draught
Call-sign	Latitude position	Hazardous cargo
Name	Longitude position	Destination
IMO number	Timestamp	Estimated time of arrival
Length	Course over ground	
Beam	Speed over ground	
Ship type	Heading	
Location of antenna	Rate of turn	

AIS messages are transmitted via radio transponders using the VHF (Very High Frequency) band to other vessels in the vicinity, as well as shore-based base stations. Satellite AIS (S-AIS) is also becoming more prevalent, where AIS signals can be received by satellites during a pass, and transmitted to ground stations (Carson-Jackson, 2012). Each AIS message is comprised of ship-related information that include static, dynamic and voyage-related information. An overview of included information can be found in Table 3.1. AIS messages are stored in historical AIS databases, that can be accessed for later use. By plotting such historical messages as a function of time, historical ship behavior can be observed. Historical AIS data have been utilized for a variety of applications in the maritime domain (Tu et al., 2017).

Studies have, however, shown that AIS messages are error prone, and may, therefore, not always be reliable (Harati-Mokhtari et al., 2007). Such errors include input errors e.g. navigation status, destination and estimated time of arrival. It has also been found that ship type, length and beam information are often erroneous. The accuracy of dynamic data will also depend on the accuracy of the equipment utilized, and may vary. As a result, one must keep in mind that use of AIS data may involve inherent error. Such anomalous data may, however, also be detected and recovered via techniques based on data analytics (Perera, 2017a).

As discussed, it is suggested in this study to utilize historical AIS data to facilitate long-range trajectory predictions to support level 3 situation awareness. An advantage of utilizing historical AIS data is that validation is inherent in the data set. By extracting test trajectories from the data, prediction algorithms can be trained on the remainder of the data, and validated against the test trajectories. The data set utilized in this study was provided by the Norwegian Coastal Administration. All AIS data available for a region surrounding the city of Tromsø, Norway from January 1st, 2017 to January 1st, 2018, were utilized. This corresponded to approximately 15 million AIS messages.

3.3 Eigendecomposition-Based Dimensionality Reduction

Dimensionality reduction facilitates the generation of low-dimensional representations of the input data. Such representations may, for instance, be more conducive to clustering algorithms. In some cases, this may be due to issues related to the curse of dimensionality (Bellman, 1961) for high dimensional data. In this study, dimensionality reduction is utilized to facilitate representations of historical AIS trajectories. The methods in this section are relevant for Papers I, II and III.

3.3.1 Principle Component Analysis

One of the most common methods of dimensionality reduction is the Karhunen-Loève (KL) transform (Karhunen, 1946), often referred to as Principle Component Analysis (PCA). PCA is a form of unsupervised learning, where the purpose is to transform data with correlated features into uncorrelated components. Such uncorrelated features may support the performance of subsequent machine learning algorithms. Given a data vector $\mathbf{x} \in \mathbb{R}^d$, where d is the dimensionality, the mean, $\boldsymbol{\mu}$ (3.1), and covariance, $\boldsymbol{\Sigma}$ (3.2), can be determined.

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}] \quad (3.1)$$

$$\boldsymbol{\Sigma} = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] \quad (3.2)$$

The eigenvectors and eigenvalues of the covariance matrix can then be calculated. Eq. (3.3) shows the relationship between $\boldsymbol{\Sigma}$ and matrix \mathbf{E} , which consists of the eigenvectors of $\boldsymbol{\Sigma}$. $\boldsymbol{\Lambda}$ is the diagonal matrix of the eigenvalues corresponding to each eigenvector.

$$\boldsymbol{\Sigma} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T \quad (3.3)$$

The covariance matrix describes the variation in the data set. By investigating its eigenvectors, one is able to identify the directions in which the data has the greatest variation. By projecting the data onto the space spanned by the eigenvectors of the covariance matrix, an uncorrelated data set can be attained. This is achieved via the KL-transform in Eq. (3.4), where \mathbf{y} is the uncorrelated feature vector.

$$\mathbf{y} = \mathbf{E}^T \mathbf{x} \quad (3.4)$$

The degree of variation is determined by the magnitude of the corresponding eigenvalues. If one orders the eigenvectors in \mathbf{E} based on the magnitude of their corresponding eigenvalues, the top eigenvectors will constitute the principal components of the data. If one projects onto the subspace spanned by the top l components, where $l < d$, one will retain most of the information in the data set, whilst also reducing the dimensionality. Such techniques are often used for visualization, where high dimensional data sets can be visualized based on the projection of the data onto the subspace spanned by the top principle components.

3.3.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) (Fisher, 1936) is a form of supervised learning that generates features with optimal separation between C different classes, each class is denoted c . Utilizing such features should, therefore, support the performance of classification. To achieve this, the scatter of the data is investigated. \mathbf{S}_m (3.5) is the mixture scatter matrix of the data, comprised of the within-class scatter matrix \mathbf{S}_w and the between-class scatter matrix \mathbf{S}_b .

$$\mathbf{S}_m = \mathbf{S}_w + \mathbf{S}_b \quad (3.5)$$

\mathbf{S}_w (3.6) describes the scatter within each class. In this sense, it describes how compact each class is. \mathbf{S}_b (3.7) describes the scatter between all classes, i.e. how spread out each class is relative to the global mean $\boldsymbol{\mu}_g$ (3.8). The prior of each class is denoted π_c .

$$\mathbf{S}_w = \sum_{c=1}^C \pi_c \boldsymbol{\Sigma}_c \quad (3.6)$$

$$\mathbf{S}_b = \sum_{c=1}^C \pi_c (\boldsymbol{\mu}_c - \boldsymbol{\mu}_g)(\boldsymbol{\mu}_c - \boldsymbol{\mu}_g)^T \quad (3.7)$$

$$\boldsymbol{\mu}_g = \sum_{c=1}^C \pi_c \boldsymbol{\mu}_c \quad (3.8)$$

In a classification setting, it is desirable for the data in each class to be as compact as possible. This can be achieved by minimizing the trace of \mathbf{S}_w . Furthermore, classification performance will be enhanced for data in which classes are spread out. In this manner, the trace of \mathbf{S}_b should be maximized. This is achieved by maximizing the class separability measure J_s (3.9).

$$J_s = \text{Tr}(\mathbf{S}_w^{-1}\mathbf{S}_m) \quad (3.9)$$

LDA is facilitated via a transformation (3.10) that maximizes J_s in the transformed space.

$$\mathbf{y} = \mathbf{A}^T \mathbf{x} \quad (3.10)$$

The optimal solution is found such that $\mathbf{A} = \mathbf{E}$, where \mathbf{E} is the matrix of eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The relationship is defined in (3.11), where $\mathbf{\Lambda}$ is the diagonal matrix of corresponding eigenvalues. The transformation is defined in (3.12), where \mathbf{x} is projected onto the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$.

$$\mathbf{S}_w^{-1}\mathbf{S}_b = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T \quad (3.11)$$

$$\mathbf{y} = \mathbf{E}^T \mathbf{x} \quad (3.12)$$

LDA may also inherently reduce the dimensionality, as \mathbf{S}_b is of rank $C - 1$. $\mathbf{S}_w^{-1}\mathbf{S}_b$ is, therefore, also of rank $C - 1$. Hence, there will be $C - 1$ non-zero eigenvalues, and $\mathbf{y} \in \mathbb{R}^d$ where $d = C - 1$. The dimensionality can also be further reduced, as in PCA, using $d < C - 1$ eigenvectors. However, optimality with respect to J_s is only preserved such that $d = C - 1$. For further details see Theodoridis & Koutroumbas (2009).

3.4 Clustering

Clustering is a form of unsupervised learning that strives to discover the underlying structure in the data, and group similar data points together. In this manner, clustering can be thought of a method to categorize data based on some similarity

measure. As such, clustering techniques emulate the development of categorization functions that support high levels of situation awareness in humans. This section will present relevant clustering techniques used in this study. First, Gaussian Mixture Models are presented, where clustering is facilitated via the Expectation Maximization algorithm. This approach is relevant for the work in Papers I, II and III. The Hierarchical Density-Based Clustering of Applications with Noise algorithm is then introduced, relevant for the work in Papers IV and V.

3.4.1 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) (Reynolds et al., 2000) can be utilized to model the underlying Probability Density Function (PDF) of a data set in (3.13). It is assumed that the data is comprised of a mixture of M Gaussian distributions, each with a mean $\boldsymbol{\mu}_m$, covariance $\boldsymbol{\Sigma}_m$, and prior probability π_m . The PDF of model m is given in Eq. (3.14), where d is the dimensionality of the data.

$$p(\mathbf{x}) = \sum_{m=1}^M \pi_m \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \quad (3.13)$$

$$\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_m|}} e^{(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m))} \quad (3.14)$$

The class membership vector, $\mathbf{v}_i \in \mathbb{R}^M$, is introduced for the i^{th} data point, \mathbf{x}_i , in (3.15). The class conditional probability is defined in (3.16).

$$\mathbf{v}_{ik} = \begin{cases} 1 & \text{if } k = m \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

$$p(\mathbf{x}_i | \mathbf{v}_{im} = 1) \sim \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \quad (3.16)$$

Assuming independence, the joint probability $p(\mathbf{x}_i, \mathbf{v}_i; \Theta)$ is given in (3.17). The most likely mixture model that fits N data points is found by maximizing the log-likelihood (LL) in (3.18) with respect to the model parameters Θ .

$$p(\mathbf{x}_i, \mathbf{v}_i; \Theta) = \prod_{m=1}^M [p(\mathbf{x}_i | \mathbf{v}_{im} = 1; \Theta) p(\mathbf{v}_{im} = 1)]^{\mathbf{v}_{im}} \quad (3.17)$$

$$LL(\Theta) = \sum_{i=1}^N \sum_{m=1}^M (\mathbf{v}_{im} \log(p(\mathbf{x}_i | \mathbf{v}_{im} = 1; \Theta)) + \mathbf{v}_{im} \log(\pi_m)) \quad (3.18)$$

Expectation Maximization Algorithm

GMM clustering via the Expectation Maximization (EM) algorithm (Dempster et al., 1977) is a parametric clustering technique. The goal of the clustering algorithm is to assign class membership to all data points, and in this manner cluster the data. To facilitate this, the log-likelihood (3.18) is maximized via the EM algorithm. The algorithm alternates between evaluating the expected class membership of data points given the current GMM (expectation step), and the maximization of model parameters given the updated class membership (maximization step). By repeating this process iteratively, it has been shown that the algorithm may converge to a local optimum (Wu, 1983).

Prior to applying the EM-algorithm, the number of underlying models, M , must be input. Model parameters will then be initialized for all distributions in the GMM. A common initialization method is to initialize all the means, $\boldsymbol{\mu}_m$, as randomly chosen data points. Furthermore, the the covariance for each Gaussian is set to the identity matrix, $\boldsymbol{\Sigma}_m = \mathbf{I}$, and the prior probabilities set equal, $\pi_m = \frac{1}{M}$. Once all models are initialized, the expectation step is conducted by evaluating the expected class membership $\langle \mathbf{v}_{im} \rangle$ in (3.19).

$$\langle \mathbf{v}_{im} \rangle = \frac{p(\mathbf{x}_i | \mathbf{v}_{im} = 1; \Theta) \pi_m}{\sum_{k=1}^M p(\mathbf{x}_i | \mathbf{v}_{ik} = 1; \Theta) \pi_k} \quad (3.19)$$

The underlying data belonging to each distribution have, therefore, changed, and the model parameters must be updated based on the updated class membership parameters in the maximization step. This is conducted by maximizing the log-likelihood (3.18). The estimated parameters for each model in the GMM are calculated in (3.20)-(3.22).

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{i=1}^N \langle \mathbf{v}_{im} \rangle \mathbf{x}_i}{\sum_{i=1}^N \langle \mathbf{v}_{im} \rangle} \quad (3.20)$$

$$\hat{\boldsymbol{\Sigma}}_m = \frac{\sum_{i=1}^N \langle \mathbf{v}_{im} \rangle (\mathbf{x}_i - \boldsymbol{\mu}_m)(\mathbf{x}_i - \boldsymbol{\mu}_m)^T}{\sum_{i=1}^N \langle \mathbf{v}_{im} \rangle} \quad (3.21)$$

$$\hat{\mu}_m = \frac{\sum_{i=1}^N \langle \mathbf{v}_{im} \rangle}{N} \quad (3.22)$$

The EM algorithm then repeats iteratively until a stopping criteria is met. This iterative process allows the GMM to adapt to the underlying data, such that the most likely PDF can be estimated. A common stopping criteria is to evaluate the log-likelihood (3.18), where convergence may indicate an optimal solution. Alternatively, convergence with respect to model parameters can be evaluated. Convergence, however, may be a challenge when utilizing the EM-algorithm. One technique to avoid divergence is to utilize a number of random starts, where different initializations of the GMM are utilized. For each initialization, the EM-algorithm is run for a number of iterations. The initialization with the highest log-likelihood is selected, and run for further iterations until convergence.

Bayesian Information Criterion

In many cases, the number of classes may not be known. As a result, the GMM is inherently constrained by the input parameter, M . However, techniques exist to estimate the most likely number of clusters. One method is to evaluate the Bayesian Information Criterion (*BIC*) (3.23) (Schwarz et al., 1978). $LL(\Theta)$ is the log-likelihood at the optimum, K the number of free parameters in the GMM, and N the number of data points.

$$BIC = -2LL(\Theta) + K \ln(N) \quad (3.23)$$

The BIC can be thought of as a measure of the log-likelihood and complexity of the model. Hence, the model with the highest log-likelihood and least complexity should be chosen. By varying the value of M , the BIC of multiple GMMs can be calculated. The most likely GMM is that with the lowest BIC value.

3.4.2 Hierarchical Density-Based Clustering of Applications with Noise

The Hierarchical Density-Based Clustering of Applications with Noise (HDBSCAN) algorithm (Campello et al., 2013) is a non-parametric clustering technique. The algorithm is an extension of the Density-Based Clustering of Applications with Noise (DBSCAN) algorithm (Ester et al., 1996). Such an approach can be advantageous compared to parametric methods as they are capable of identifying

clusters of varying density and shape. Furthermore, the algorithm is capable of discovering the most likely number of clusters automatically.

The algorithm calculates local density estimates based on core distance measures. The core distance of each point, D_c , is defined as the distance to the k^{th} -nearest neighbor, and indicates the density of the neighborhood. HDBSCAN then calculates the mutual reachability distance, D_m , between two points \mathbf{x}_i and \mathbf{x}_j in (3.24).

$$D_m(\mathbf{x}_i, \mathbf{x}_j) = \max(D_c(\mathbf{x}_i), D_c(\mathbf{x}_j), \|\mathbf{x}_i - \mathbf{x}_j\|_2) \quad (3.24)$$

HDBSCAN subsequently constructs a minimum spanning tree based on the mutual reachability distance, which is converted into a hierarchy of connected components. A minimum cluster size is input by the user as a hyper-parameter. Any clusters that are below the minimum cluster size are filtered out. The most stable remaining clusters in the hierarchy are then selected. However, no cluster that is a descendant of a previously selected cluster in the hierarchy may be chosen. Any data points not belonging to the selected clusters are defined as noise. For further details on the algorithm, see Campello et al. (2013).

3.5 Anomaly Detection

Anomaly detection (Chandola et al., 2009) is a widely researched field. One application of anomaly detection in the maritime domain is to detect anomalous ship behavior (Riveiro et al., 2018). Anomaly detection is generally parametric or non-parametric. In this thesis, parametric anomaly detection based on fitted Gaussian distributions (3.14) is applied to clusters of historical ship trajectories.

As such, a Gaussian distribution is fit to representations of the historical trajectories, where the most common behavior will lie close to the mean of the distribution. Abnormal ship behavior should, therefore, lie far from the mean. By evaluating the standard deviation contours of the distribution, anomalies can be detected based on thresholding. In this sense, any data points within the contours are defined as normal behavior, and any outside are defined as anomalies.

3.6 Deep Learning

Deep learning is a sub-field of machine learning that has gained much attention in recent years due to its state-of-the-art performance in a variety of domains e.g. computer vision (Voulodimos et al., 2018) and natural language processing (Cho et al., 2014). Deep learning leverages artificial neural networks that are optimized via the back-propagation algorithm (Rumelhart et al., 1986). This optimization is often referred to as training, where a loss function is evaluated, and the errors back-propagated through the network to calculate the gradients. Gradient descent is then applied to update the model parameters. Some of the most popular methods include AdaGrad (Duchi et al., 2011) and Adam (Kingma & Ba, 2015). In this manner, the network is able to learn the model parameters that minimize the loss function.

The training is run over a number of iterations, where predictions are compared to the labels. Once the loss has converged, the training is stopped. Such recurrent comparisons of predictions to labels is mirrored in the manner in which humans develop high level situation awareness. In the case of humans, however, predictions are validated against observations, and recurrently updated based on new experience. For the case of deep learning, such experience is facilitated by data. The amount of experience a human has should enhance their predictive abilities. The same applies to deep learning, where a model trained on few data points will not perform well. As a result, deep learning requires large amounts of data. For the purpose of this study, it was found that historical AIS data provides a data set of sufficient size to support such techniques. In this section, selected deep learning techniques relevant for supporting level 3 situation awareness are presented. These are relevant for Paper II, IV and V.

3.6.1 Multi-Layer Perceptron

A feed forward neural network, or Multi-Layer Perceptron (MLP), is the the most basic form of deep learning. In essence, an MLP is a mathematical function that maps the input to some output, via a number of hidden representations. Such hidden representations are learned during the training of the network, and aid in achieving the desired output. This is facilitated by connected neurons, where each neuron is a perceptron.

The McCulloch-Pitts neuron (McCulloch & Pitts, 1943) was developed to imitate brain functions, specifically the firing of neurons. This was further developed in the perceptron (Rosenblatt, 1958). The perceptron utilized in modern neural networks

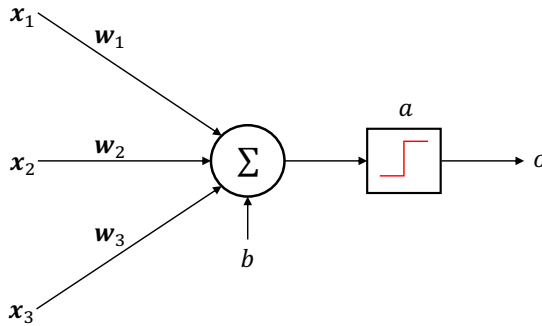


Figure 3.1: Perceptron.

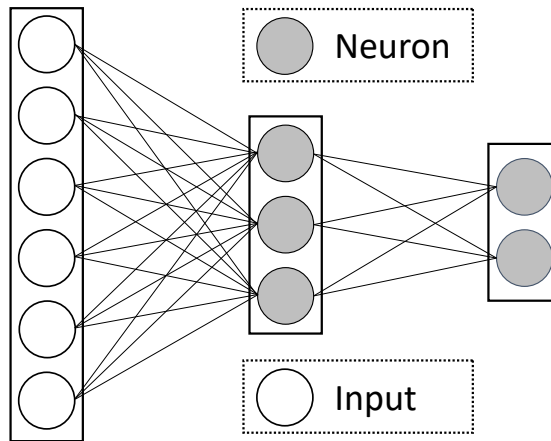


Figure 3.2: Multi-Layer Perceptron. Each neuron is a perceptron.

applies a set of weights, \mathbf{w} , to an input vector, \mathbf{x} , and takes the sum along with a bias term b . This is then run through an activation function, a , which generates the output, o . The operation is calculated in (3.25), illustrated in Fig. 3.1.

$$o = a(\mathbf{w}^T \mathbf{x} + b) \quad (3.25)$$

By stacking multiple perceptions, a layer of perceptrons is created. An MLP utilizes multiple layers, where the output of each is the input to the next. This is the basis for a neural network that consists of fully connected neurons, i.e. perceptrons. As such, MLP layers are often referred to as fully connected layers. An MLP is illustrated in Fig. 3.2.

By utilizing multiple layers, the network can be made deeper. For each layer, the network conducts a transformation of the data to a hidden representation. Non-linear activation functions are often used to introduce non-linearity to the network. Common activation functions include the Sigmoid (3.26) and Tanh (3.27) activation functions. More recently, the Rectified Linear Unit (ReLU) (3.28) (Nair & Hinton, 2010) has become popular due to its ability to allow gradients to flow more effectively in deep networks.

$$a(s) = \frac{1}{1 + e^{-s}} \quad (3.26)$$

$$a(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \quad (3.27)$$

$$a(s) = \begin{cases} 0 & \text{if } s \leq 0 \\ s & \text{if } s > 0 \end{cases} \quad (3.28)$$

3.6.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986) are designed to process sequences, and are, therefore, ideal for processing time series data e.g. AIS trajectories. RNNs are able to incorporate temporal dependencies, and can in this sense be thought of as having a form of memory. RNNs learn what information is important to retain, and what to forget with respect to achieving a specific goal. They can be viewed as an unfolded computational graph, where the same operation is applied at all time steps. In this sense, the operation is recurrent.

Given a time series $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_L\}$, an RNN processes each input state \mathbf{x}_t sequentially. The memory of the network is incorporated in the network through a hidden state \mathbf{h}_t . The recurrent operation, f , is calculated in (3.29), where the previous hidden state \mathbf{h}_{t-1} and current input state \mathbf{x}_t are processed using the same RNN cell. As a result, the RNN parameters, Θ , are shared across all operations. The operation outputs the current hidden state \mathbf{h}_t , which is fed to the next step. This is illustrated in Fig. 3.3, with the recurrent operation in red. In this manner, all relevant past information is stored in the hidden state, and passed further down the network.

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t; \Theta) \quad (3.29)$$

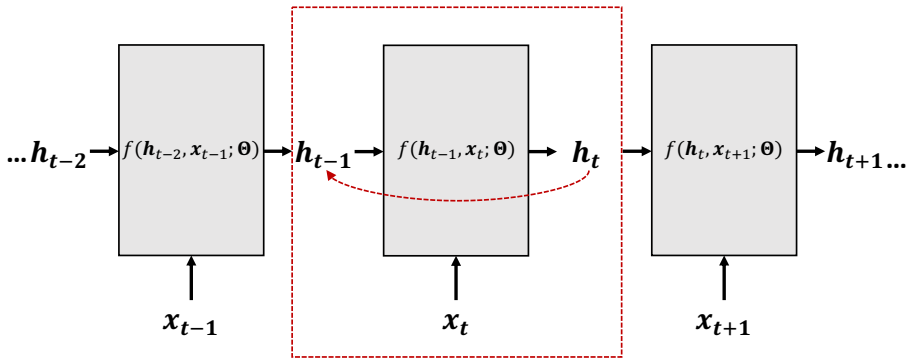


Figure 3.3: RNN. Figure from Paper V.

The standard RNN operates in the forward direction. In this manner, temporal dependencies will only be calculated from the past towards the future. Schuster & Paliwal (1997) introduced the bidirectional RNN, which processes sequences in both the forward, and backward directions concurrently. In this manner, the past is viewed as dependent on the future. In the case of AIS data, past behavior may be affected by future maneuvers (e.g. reducing speed before changing course), and may, therefore, add information to better describe the trajectories. Furthermore, the outlined architecture is only one layer. In Graves et al. (2013) it was found that increasing the depth of an RNN increased the performance. Such multi-layer RNNs are referred to as stacked RNNs, in that multiple RNNs are stacked on top on one another. Each feeds into the next as illustrated in Fig. 3.4.

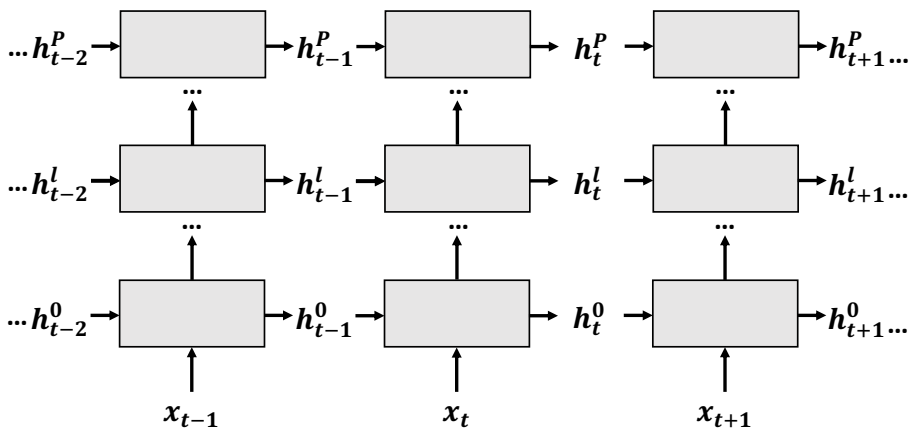


Figure 3.4: P -layer stacked RNN. Each layer is denoted l . Figure from Paper V.

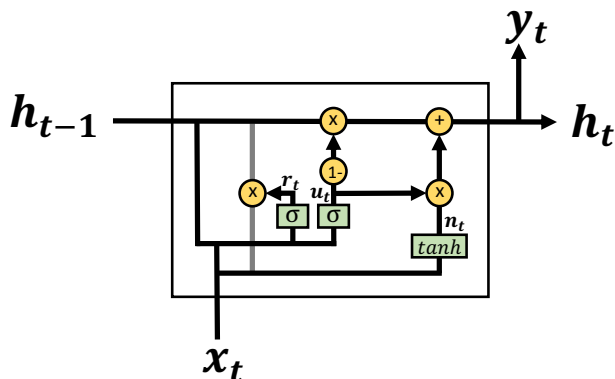


Figure 3.5: Gated Recurrent Unit architecture. Figure from Paper IV.

Gated Recurrent Unit

The original architecture of the RNN, however, struggles to learn long-term dependencies in the data due to the challenge of vanishing gradients during back-propagation (Bengio et al., 1994). As a result, more advanced architectures have been developed to ameliorate this issue. It was found that introducing gates to allow, or restrict the flow of data aided the ability of the networks to learn long-term dependencies. Such architectures include the Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), and the more recent Gated Recurrent Unit (GRU) (Cho et al., 2014; Chung et al., 2014).

The GRU is closely related to the popular LSTM architecture, but reduces the number of learned parameters. Hence, the training time of the GRU is reduced compared to the LSTM. The GRU architecture is illustrated in Fig. 3.5, where one GRU cell is visualized. As such, the internal mechanisms of the cells in Fig. 3.3 and Fig. 3.4, can be thought of as consisting of GRUs.

To regulate the flow of information, the GRU utilizes a reset gate, \mathbf{r}_t , (3.30) and update gate, \mathbf{u}_t , (3.31). These gates learn what information should be passed on to the next cell, i.e. what should be remembered. The parameters in each gate are comprised of weight matrices, \mathbf{W} , and bias vectors, \mathbf{b} . They can, therefore, be thought of as two fully connected layers that operate on the input, \mathbf{x}_t , and previous hidden state, \mathbf{h}_{t-1} , respectively. The sum of the outputs are run through Sigmoid activation functions (3.26), denoted σ . As a result, the output will be constricted between values of 0 and 1. A value of 1 entails an open gate, and 0 a closed gate. Hence, the flow of information can be regulated by applying the Hadamard product of \mathbf{r}_t and \mathbf{u}_t with the respective inputs.

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{b}_{xr} + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r) \quad (3.30)$$

$$\mathbf{u}_t = \sigma(\mathbf{W}_{xu}\mathbf{x}_t + \mathbf{b}_{xu} + \mathbf{W}_{hu}\mathbf{h}_{t-1} + \mathbf{b}_{hu}) \quad (3.31)$$

In (3.32), a new candidate vector, \mathbf{n}_t is calculated. The current hidden state, \mathbf{h}_t , is calculated in (3.33). As such, all the relevant information of the past sequence is stored in the hidden state, and passed on to the next cell. (3.29), can therefore be substituted with (3.33) when utilizing a GRU. A prediction at each state, \mathbf{y}_t , can also be calculated via a fully connected layer in (3.34). For a more detailed explanation of the GRU, see Cho et al. (2014); Chung et al. (2014)

$$\mathbf{n}_t = \tanh(\mathbf{W}_{xn}\mathbf{x}_t + \mathbf{b}_{xn} + \mathbf{r}_t \odot (\mathbf{W}_{hn}\mathbf{h}_{t-1} + \mathbf{b}_{hn})) \quad (3.32)$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \mathbf{n}_t \quad (3.33)$$

$$\mathbf{y}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_{hy} \quad (3.34)$$

3.6.3 Autoencoders

The objective of many deep learning architectures is often classification or regression. However, deep learning can also be applied in an unsupervised manner. In this case, deep learning facilitates meaningful representations of the data. Such representations may reveal the underlying structure of the data, that may not be readily apparent in the input space. Furthermore, the dimensionality of the data may be high, constituting a challenge for standard clustering algorithms. Hence, low dimensional representations should be generated. One method to facilitate such representations is via autoencoder architectures.

Linear Autoencoder

The simplest form of an autoencoder is an MLP (Bourlard & Kamp, 1988), illustrated in Fig. 3.6. An autoencoder is comprised of two parts: an encoder (3.35), and a decoder (3.36).

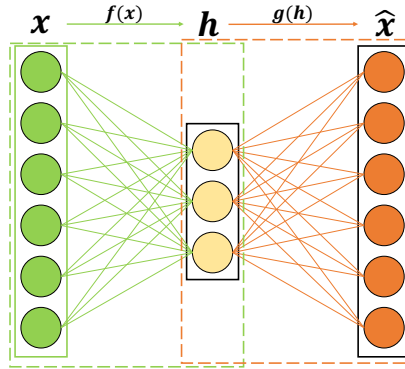


Figure 3.6: Linear autoencoder. The encoder is illustrated in green, and the decoder in orange. Figure adapted from Paper II.

$$\mathbf{h} = f(\mathbf{x}) \quad (3.35)$$

$$\hat{\mathbf{x}} = g(\mathbf{h}) \quad (3.36)$$

The encoder produces the code, \mathbf{h} , which is a representation of the input space. The decoder then reconstructs the input from the code. One method to facilitate meaningful representations of the data is to utilize an undercomplete autoencoder structure (Goodfellow et al., 2016). This entails that the code, \mathbf{h} , has a lower dimensionality than the input space, as illustrated in Fig. 3.6. As a result, this implies a compression of the data, i.e. dimensionality reduction. If the autoencoder is trained using the reconstruction loss, the encoder will learn to generate meaningful representations of the data, that preserve as much mutual information between the code and input space as possible.

MLPs generally have non-linear activation functions. However, if one does not utilize such activation functions, the encoder and decoder will simply be linear transformations of the data, i.e a linear autoencoder. If one considers a 2-layer linear autoencoder as illustrated in Fig. 3.6, the encoder can be defined according to (3.37), and the decoder according to (3.38). (3.39) calculates the mean squared error loss, J . By training the network with this loss function, the minimum reconstruction has been shown to be found such that $\mathbf{A} = \mathbf{W}$, where the columns of \mathbf{W} span the orthonormal basis spanned by the eigenvectors of the covariance matrix of the data, and $\mathbf{c} = \boldsymbol{\mu}$ (Goodfellow et al., 2016). The encoder is, therefore, analogous to PCA in this case, where (3.37) is equal to the KL-transform in (3.4).

$$\mathbf{h} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}) \quad (3.37)$$

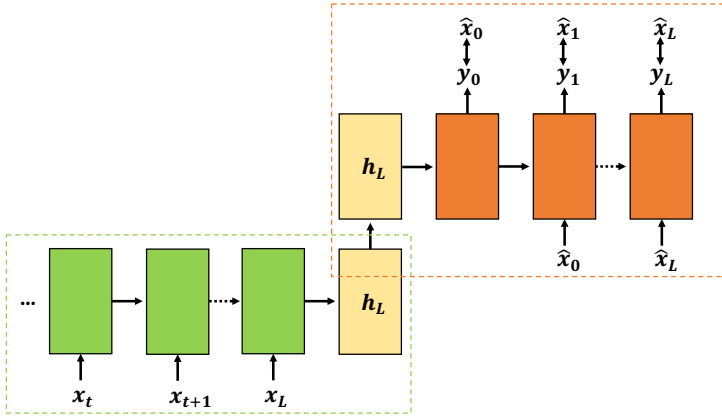


Figure 3.7: Recurrent autoencoder. The encoder is illustrated in green, and the decoder in orange. Figure from Paper IV.

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{h} + \mathbf{c} \quad (3.38)$$

$$J = \mathbb{E} [|\mathbf{x} - \hat{\mathbf{x}}|^2] \quad (3.39)$$

Recurrent Autoencoder

Sequence to sequence models (Sutskever et al., 2014) are one of the most popular forms of RNNs, and form the basis for many natural language processing tasks e.g. translation (Cho et al., 2014). For such tasks, an input sequence, e.g. a sentence in English, is run through an encoder RNN. A decoder RNN then predicts a target sequence, e.g. the corresponding sentence in Spanish. By applying this encoder-decoder architecture to reconstruct the input instead of a target sequence, the model functions as a Recurrent Autoencoder (RAE) (Srivastava et al., 2015).

An RAE architecture is illustrated in Fig. 3.7. The encoder is capable of processing variable length time series, e.g. AIS trajectories, and compressing them into a fixed size vector, \mathbf{h}_L , i.e. the final hidden state in the encoder network. The decoder network takes \mathbf{h}_L , and must reconstruct the input from this vector sequentially, where the the next state is estimated according to (3.34).

The network must learn to retain as much mutual information as possible between the input sequence and its compressed representation, \mathbf{h}_L . Utilizing such an architecture, sequences of variable length can be encoded to equally sized representations of lower dimensionality. To reconstruct the input as accurately as possible, similar data points will be grouped more closely in the representation space. Such

learned representations should, therefore, provide a better foundation for a subsequent clustering algorithm.

Variational Recurrent Autoencoder

The Variational Autoencoder (VAE) (Kingma & Welling, 2014; Rezende et al., 2014) is a probabilistic approach to the autoencoder, that assumes that data are generated from a continuous latent variable, \mathbf{z} . The encoder side of the VAE produces a distribution over the latent variable via $q_\phi(\mathbf{z}|\mathbf{x})$. This function is further assumed to be a multivariate Gaussian with a diagonal covariance in (3.40). The decoder, $p_\theta(\mathbf{x}|\mathbf{z})$, reconstructs \mathbf{x} from \mathbf{z} .

$$q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2 \mathbf{I}) \quad (3.40)$$

Standard autoencoders that are trained using the reconstruction loss, will strive to optimally utilize the code space. As a result, this often leads to it being sparsely populated (Spinner et al., 2018). Such sparse representations may not be conducive to clustering algorithms. One of the advantages of utilizing a VAE compared to a standard autoencoder, is that it encourages the latent variables to become normally distributed. As a result, more compact groupings of data are encouraged, limiting the chaos in the latent space. Such representations will likely provide a better basis for a clustering algorithm.

Fabius & van Amersfoort (2015) introduced the Variational Recurrent Autoencoder (VRAE), where the encoder and decoder are approximated by RNNs, illustrated in Fig. 3.8. The output of the encoder RNN is the final hidden state \mathbf{h}_L . This encodes the entire input sequence. Subsequently, the mean $\boldsymbol{\mu}_z$ and standard deviation $\boldsymbol{\sigma}_z$ in (3.40) are estimated using fully connected layers in (3.41) and (3.42).

$$\boldsymbol{\mu}_z = \mathbf{W}_\mu \mathbf{h}_L + \mathbf{b}_\mu \quad (3.41)$$

$$\boldsymbol{\sigma}_z = \mathbf{W}_\sigma \mathbf{h}_L + \mathbf{b}_\sigma \quad (3.42)$$

The latent variable, \mathbf{z} , is then sampled from the Gaussian distribution in (3.40). However, back-propagation is not possible through such a sampling operation. As a result, the re-parametrization trick is utilized in (3.43). $\boldsymbol{\epsilon}$ is sampled from a Gaussian distribution such that $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. In this manner, gradients can flow freely irrespective of the sampling operation.

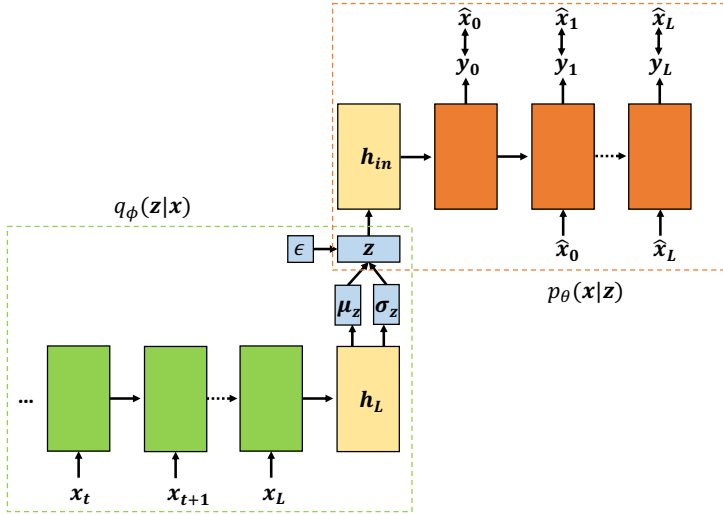


Figure 3.8: Variational recurrent autoencoder. The encoder is illustrated in green, and the decoder in orange. Figure adapted from Paper IV.

$$\mathbf{z} = \boldsymbol{\mu}_z + \boldsymbol{\sigma}_z \odot \boldsymbol{\varepsilon} \quad (3.43)$$

As for the standard RAE, the decoder RNN takes a hidden state as input. This initial hidden state, \mathbf{h}_{in} , is estimated using a fully connected layer in (3.44). The decoder then reconstructs the input as for a standard RAE.

$$\mathbf{h}_{in} = \tanh(\mathbf{W}_{zh}\mathbf{z} + \mathbf{b}_{zh}) \quad (3.44)$$

The network is optimized by maximizing a variational lower bound on the log-likelihood. This is facilitated via the loss function, J , in (3.45). The first term can be maximized by minimizing the reconstruction loss, e.g. via the mean squared error (3.39). The second term is the negative Kullback-Leibler divergence (D_{KL}) between the encoder, $q_\phi(\mathbf{z}|\mathbf{x})$, and the prior $p_\theta(\mathbf{z})$. It is further assumed that the prior is normally distributed according to $p_\theta(\mathbf{z}) \sim \mathcal{N}(0, \mathbf{I})$. By minimizing the Kullback-Leibler divergence, one maximizes the second term. As such, the network is encouraged to learn compact groupings of data in the latent space. For further details on the VAE and VRAE, see Kingma & Welling (2014); Rezende et al. (2014); Fabius & van Amersfoort (2015).

$$J(\theta, \phi; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log(p_\theta(\mathbf{x}|\mathbf{z}))] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})) \quad (3.45)$$

Part II

Research Outcome

Chapter 4

Summary of Research

In this chapter, the included publications are summarized. The papers are presented in an order that reflects the progression of the research. In the first section, papers that facilitate local ship behavior prediction are presented. Here Paper I, II and III are summarized. These approaches aim to facilitate live predictions based on locally extracted trajectory segments. The next section presents the work on applying deep learning techniques to facilitate regional ship behavior prediction. Paper IV, and V are summarized in this section. To improve the readability of this thesis, figures have been included from the appended papers. For further details on the figures, please see the referenced papers.

4.1 Local Ship Behavior Prediction

In this section, methods supported by locally extracted trajectories are investigated. This entails that historical AIS trajectories are extracted relative to the current position, course over ground, and speed over ground of a selected vessel. The duration of these trajectories is defined based on the desired prediction horizon, which in this thesis is suggested to be 30 minutes, as discussed in Sec. 2.3.2. As such, the extracted future 30 minute trajectories of similar historical vessels are assumed to represent the distribution of possible future behavior for the selected vessel. The trajectories contain the spatial data, as well as the course over ground and speed over ground data.

Such local methods, however, must be conducted live, as the trajectories are extracted based on the current state of the selected vessel. As such, pre-trained mod-

els cannot be utilized, as the data used to train the models vary for each prediction. Efficient methods must, therefore, be utilized. As a result, machine learning techniques using eigenvector-based approaches are investigated in the following papers.

Paper I - Ship Behavior Prediction via Trajectory Extraction-Based Clustering for Maritime Situation Awareness

Paper I presents an approach to ship behavior prediction via trajectory extraction-based clustering. The objective of the paper is to cluster historical AIS trajectory segments via low-dimensional representations. Using these clusters, the future ship trajectory is predicted after classification to one of the clusters. As such, this paper addresses RO1, RO2 and RO3, with the primary contribution towards RO3 (i). The research objectives are repeated below to improve the readability of the thesis:

RO1 Leverage machine learning to provide methods to support maritime situation awareness.

RO2 Leverage historical AIS data to provide methods to support level 3 maritime situation awareness by artificially serving as navigational experience.

RO3 Develop methods to emulate the development of high level maritime situation awareness in humans by:

- (i) Developing methods to categorize ship behavior.
- (ii) Developing methods to facilitate behavior models for predicting future ship behavior.
- (iii) Developing methods to facilitate pattern matching of observed ship behavior.

An initial version of this paper was presented in Murray & Perera (2019). Murray & Perera (2019) extended the work in Murray & Perera (2018), where it was found that extracting a subset of historical AIS data, relative to the current state of a selected vessel, enhanced the accuracy of a data-driven trajectory prediction technique developed in Hexeberg et al. (2017). Both methods were found in Murray & Perera (2018) to have superior performance to the constant velocity method commonly used in collision avoidance settings. However, the performance was degraded in cases where the underlying data diverged (e.g. multiple ship routes).

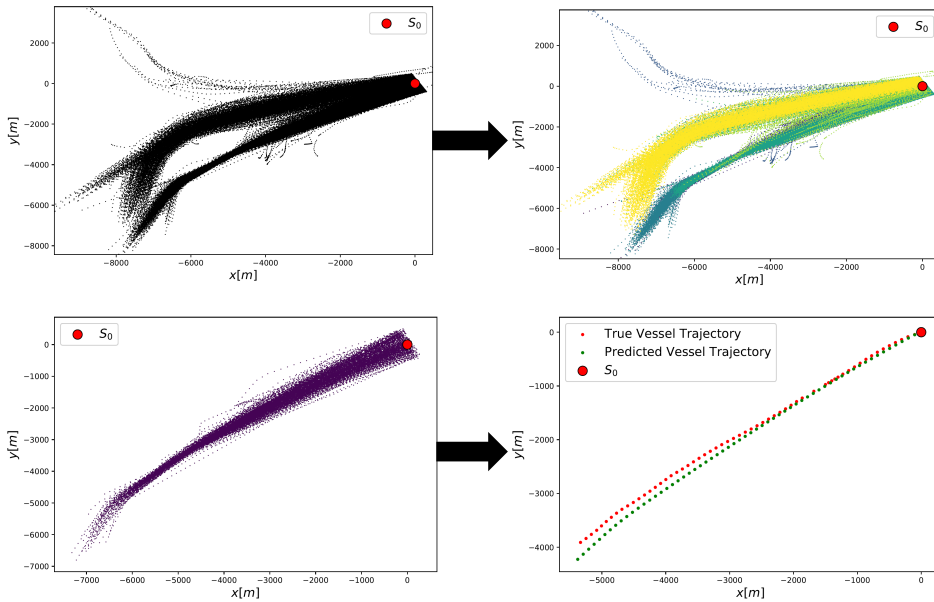


Figure 4.1: The extracted trajectories (top left) and clusters of trajectories (top right), along with the classified cluster (bottom left) and predicted future trajectory (bottom right). Figures from Paper I.

The method in Paper I extracts relevant historical AIS trajectory segments to represent the possible future 30 minute behavior of a selected vessel. Low dimensional representations of the extracted trajectory segments are then generated using the KL-transform. Gaussian Mixture Models are then employed to cluster the trajectory representations. The approach was found to be successful in discovering clusters of specific ship behavior. Additionally, the resultant clusters were found to be physically meaningful, where they appeared to represent ship routes, as well as speed clusters within various routes. By clustering 30 minute trajectory segments, clusters of specific behavior are discovered. If the trajectories for an entire region were utilized, such specific behavior would likely be ignored, resulting in larger clusters of more general behavior. Elements of the method are illustrated in Fig. 4.1.

Next, the method extracts the past 10 minutes of historical trajectory behavior relative to the current state of the vessel. Labels for each trajectory segment are available from the clustering of their corresponding future 30 minute trajectory segments. Linear Discriminant Analysis is then employed to generate representations of the past trajectories with optimal separation between classes. It is further assumed that the past 10 minutes of the selected vessel are available, and this trajectory is projected via Linear Discriminant Analysis onto the same subspace.

The selected vessel is then classified to one of the clusters of ship behavior in this subspace.

Finally, a trajectory prediction is facilitated by applying the Single Point Neighbor Search Method (Hexeberg et al., 2017) to the subset of data in the cluster. The results show good performance, with a median error of approximately 4 % of the distance traveled for a 30 minute prediction, given accurate classification. The results, therefore, indicate that identifying clusters of specific behavior can aid in enhancing the performance of ship trajectory predictions.

Contributions by the Author

- The author conceived the ideas.
- The author developed the methodology.
- The author developed the implementation and ran all experiments.
- The author wrote the first draft of the manuscript.

Paper II - A Dual Linear Autoencoder Approach for Vessel Trajectory Prediction Using Historical AIS Data

Paper II builds upon the work in Paper I. A similar clustering and classification technique as in Paper I is applied, but the trajectory prediction is facilitated via a novel dual linear autoencoder approach. As such, this paper primarily contributes to RO3 (ii), but in so doing addresses RO1 and RO2. This approach is closely tied to the trajectory extraction and clustering techniques, as they provide the foundation for the prediction approach.

The methodology starts in a similar manner to Paper I, where relevant trajectory segments of the future 30 minute, and past 10 minute ship behavior in the AIS data set are extracted. The future trajectory segments are then clustered, and a selected vessel classified to one of the clusters in a similar manner to Paper I.

The trajectory prediction is facilitated by two linear autoencoders. A forward autoencoder is trained on the extracted future 30 minute trajectories in the relevant cluster. By encoding the data, a distribution of the ship behavior in the cluster is generated in the code-space. The future trajectory, therefore, should also belong to this distribution. As such, the decoder can be utilized to generate a new trajectory

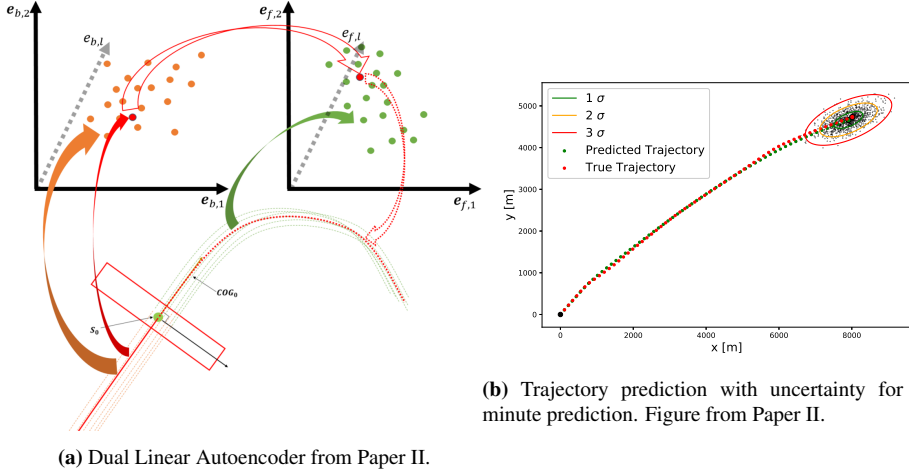


Figure 4.2: Examples of encounter situations.

by sampling from this distribution and decoding it. This is illustrated in Fig. 4.2a in green.

However, the latent representation of the future trajectory of the selected vessel is unknown. It is, therefore, estimated using the backward autoencoder. In the same manner as the forward autoencoder, the backward autoencoder is trained on the past 10 minute trajectory segments. This is illustrated in Fig. 4.2a in orange. Assuming that the past 10 minute trajectory of the selected vessel is available, it is encoded via the backward autoencoder, illustrated in red in Fig. 4.2a. A similarity measure is then evaluated between the representations generated by the autoencoder of the training data (i.e. the past trajectory segments in the cluster), and the representation of the selected vessel.

The similarity measure is utilized to estimate a distribution in the latent space of the forward autoencoder. By sampling from this distribution, the decoder generates novel trajectories that belong to the distribution. As such, a distribution of possible future trajectories is created. Uncertainty measures with respect to the future position can then be estimated based on the distribution of the future trajectories. This is illustrated for a test case in Fig. 4.2b.

The results showed superior performance to those in Paper II, with a median error of 1.6 % of the distance traveled for a prediction horizon of 30 minutes. Furthermore, the uncertainty measures were able to capture the true position the vessel in most cases. As the predictions are intended to be conducted live, the running time of the method was also investigated. On a standard laptop, it was found that the sum of the clustering, classification and prediction phases took less than four

seconds for all cases, with most lying between one and two seconds. This was considered acceptable for the purpose of the study.

Contributions by the Author

- The author conceived the ideas.
- The author developed the methodology.
- The author developed the implementation and ran all experiments.
- The author wrote the first draft of the manuscript.

Paper III - Unsupervised Trajectory Anomaly Detection for Situation Awareness in Maritime Navigation

The approaches in Paper I and Paper II identify clusters of ship behavior. Trajectory predictions are then facilitated by modeling the specific behavior in the relevant cluster. However, anomalies may degrade the performance. As a result, it is of interest to identify anomalous ship behavior, and make the user aware of the anomaly, and the possible degradation of the model with respect to predicting the future behavior of the vessel. Furthermore, the existence of anomalous trajectories in the training data will likely degrade the performance of the prediction methods. It is, therefore, of interest to identify and remove such anomalies from the historical AIS data via machine learning.

Paper III contributes towards RO1 and RO2. It is likely that humans also identify anomalous behavior when attaining situation awareness. In this manner, the paper contributes towards RO3 (iii), where future anomalies can be identified via pattern matching.

Paper III presents a method that utilizes Gaussian Mixture Model clustering to model the behavior of extracted trajectories. The trajectory extraction is facilitated in the same manner as in Paper I and Paper II. Each cluster represents a mode of ship behavior. By utilizing the distributions from the Gaussian Mixture Model, clusters of anomalous behavior can be identified directly in an intermediate anomaly detection stage. The remaining clusters should, therefore, describe normal behavior. This step is illustrated in Fig. 4.3.

Within each of the clusters of normal behavior, however, there exist local anomalies. These are discovered via three different approaches: top eigenvector analy-

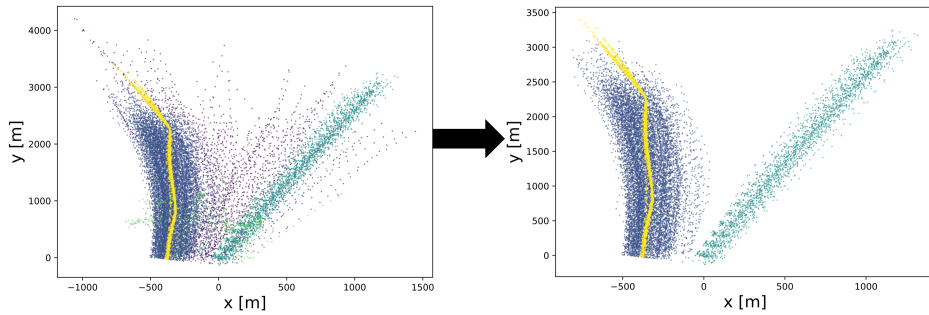


Figure 4.3: Intermediate anomaly detection and removal. Figures from Paper III.

sis, bottom eigenvector analysis and Mahalanobis distance analysis. The top and bottom eigenvector-based approaches project the trajectory segments onto the subspace spanned by the corresponding eigenvectors of the covariance matrix of the data in each cluster. Gaussian distributions are then fit to the projections, and anomaly detection is facilitated via thresholding based on the standard deviation contours.

The components of the trajectories that contributed to anomaly classification were also investigated, as it was found that the two approaches identified different anomaly modes. The top eigenvectors describe the greatest variation in the data. As such, it was found that these eigenvectors focused on sections of the trajectories that fell outside the densest regions, as well as ships that had higher speeds than normal. The bottom eigenvectors, however, describe the least variation in the data, and focused on irregular sub-trajectory segments. Finally, the Mahalanobis distance was investigated, where a combination of anomalies from the top and bottom eigenvector analysis were identified. Overall, the methods were successful in identifying various modes of anomalous ship behavior.

Contributions by the Author

- The author conceived the ideas along with the second author.
- The author developed the methodology.
- The author developed the implementation and ran all experiments.
- The author wrote the first draft of the manuscript.

4.2 Regional Ship Behavior Prediction

In this section, methods supported by regional ship behavior prediction are investigated. As opposed to the previous section, these models operate on trajectories for an entire region, and not on locally extracted trajectory segments. In this manner, models can be pre-trained for various regions using the outlined methods. An operator can then load the model for a given region to facilitate predictions.

Given that such an approach involves pre-trained models, and not run live, deep learning approaches were investigated. Due to the sequential nature of the historical AIS trajectories, recurrent architectures were investigated for use.

Paper IV - Deep Representation Learning-Based Vessel Trajectory Clustering for Situation Awareness in Ship Navigation

Paper IV investigates how deep representation learning can be leveraged to facilitate regional trajectory clustering. As such, the primary contribution of the paper is towards RO3 (i). It additionally contributes towards RO1 and RO2.

In this study, a Recurrent Autoencoder and β -Variational Recurrent Autoencoder were investigated to generate meaningful representations of the trajectories in a region. The historical AIS trajectories are multivariate time series containing the position, course over ground and speed over ground data at each time step. Furthermore, the trajectories are of variable length.

Recurrent Autoencoder architectures are capable of compressing the trajectories to fixed size vectors, irrespective of the length. As such, trajectories of various lengths can be clustered via such latent representations. The study found that both the Recurrent Autoencoder and β -Variational Recurrent Autoencoder were successful in generating meaningful representations of the AIS trajectories, where the underlying structure of the data was revealed. The results indicated, however, that the β -Variational Autoencoder generated more compact groupings of data.

To cluster the trajectories, the Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm was applied to the latent representations generated by the autoencoders. The technique was successful in identifying an unspecified number of clusters of variable shape and size that parametric methods, e.g. Gaussian Mixture Models, would likely not have been able to identify due to the constraints of the parametric descriptions. The clusters were visualized on a map, and appeared to be physically meaningful, as seen in Fig. 4.4. However, the re-

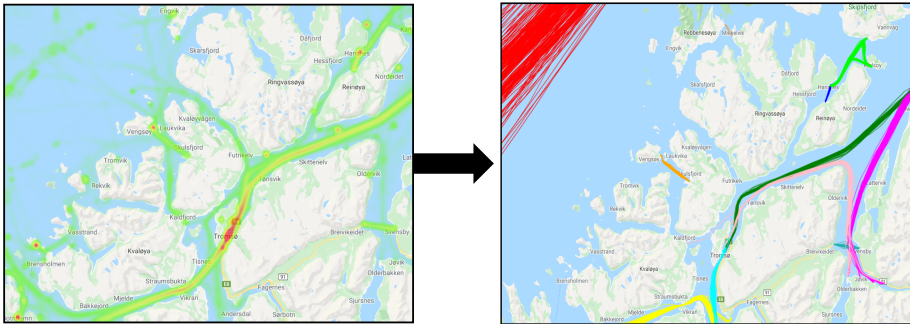


Figure 4.4: Regional trajectory clustering. Figures from Paper IV.

sults indicated that the β -Variational Autoencoder was capable of generating more compact representations than the standard Recurrent Autoencoder, supporting the clustering technique.

Contributions by the Author

- The author conceived the ideas.
- The author developed the methodology.
- The author developed the implementation and ran all experiments.
- The author wrote the first draft of the manuscript.

Paper V - An AIS-Based Deep Learning Framework for Regional Ship Behavior Prediction

Paper V builds upon the work in Paper IV to develop a deep learning framework for regional ship behavior prediction. As such, it contributes to RO1, RO2 and RO3 (i). Furthermore, it contributes especially to RO3 (ii) and RO3 (iii).

In this study, an approach similar to that in Paper I and Paper II is applied. First, trajectory representations are generated and clustered, where each cluster represents a mode of specific ship behavior. Subsequently, a selected vessel is classified to one of the clusters. A local model for the specific behavior in the cluster is then utilized to predict the future trajectory of the vessel.

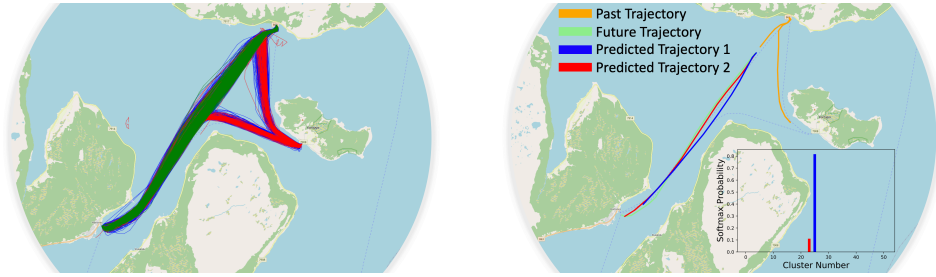


Figure 4.5: Relevant trajectory clusters (left), and resultant predictions (right) for a test case. Figures from Paper V.

The methodology is separated into two primary modules: the clustering and prediction modules. In the clustering module, a bidirectional, stacked Variational Recurrent Autoencoder is trained on the historical AIS trajectories for a region. The representations generated by running a forward pass of the encoder allow for the underlying structure to be revealed. The Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm was then applied to cluster the representations. In this manner, clusters of ship behavior were identified.

The prediction module is comprised of two sub-modules: the classification and local behavior modules. The classification module is responsible for classifying the trajectory of a selected vessel to one of the clusters, and the local behavior module for predicting the future behavior in a specific cluster. The prediction module utilizes a sequence-to-sequence model with attention to facilitate the predictions. A local behavior model is trained for each cluster present in the data set. As such, there are as many local behavior models as there are clusters.

When conducting a prediction, a selected vessel is classified to one of the clusters, and the corresponding model predicts the future trajectory. However, an uncertainty measure is introduced in the classification module. Instead of outputting one single cluster, a distribution over all clusters, i.e. classes, is generated. The framework, therefore, identifies the most likely clusters the vessel belongs to. Using the corresponding local behavior models for these clusters, multiple possible future trajectories are generated. A test case is illustrated in Fig. 4.5. The framework was applied to a test region with good results. Once trained, the framework is capable of predicting a future trajectory in under a second.

Contributions by the Author

- The author conceived the ideas.

- The author developed the methodology.
- The author developed the implementation and ran all experiments.
- The author wrote the first draft of the manuscript.

4.3 Research Contributions

To improve the readability of this thesis, the research presented in this chapter is consolidated in five research contributions, as introduced in Sec. 1.3:

RC1 Methods to generate representations of historical ship behavior

RC2 Methods to cluster historical ship behavior

RC3 Methods to classify a novel ship trajectory to a cluster of historical behavior

RC4 Methods to predict future ship behavior

RC5 Methods to identify anomalous ship behavior

The contributions of the various papers to these are summarized in Table 4.1. The level of contribution is indicated by using the "+" symbol, where "+++" indicates the highest contribution, and blank limited to no contribution. Paper III is the only paper that directly contributes to RC5. However, the methods developed in Paper IV and Paper V can indirectly be utilized to facilitate anomaly detection via HDBSCAN, but are not addressed in the studies. As such, they were assigned a minor contribution towards RC5. Furthermore, papers which leverage techniques introduced in other papers are assigned minor contributions.

Table 4.1: Paper contributions.

	Paper I	Paper II	Paper III	Paper IV	Paper V
RC1	+++	++	++	+++	++
RC2	+++	+	+	+++	++
RC3	+++	+			+++
RC4	++	+++			+++
RC5			+++	+	+

Chapter 5

Discussion

In this chapter, a discussion of the research outcome is presented. In the first section, the connection between the research contributions and objectives is discussed. The following section provides a general discussion of the research in light of the research objectives. In the first subsection, the facilitation of level 3 situation awareness is discussed along with relevant limitations. Subsequently, the use of historical AIS data is discussed along with limitations of the data. Next, machine learning, and how it was leveraged to emulate high level situation awareness of ship navigators, is discussed. Limitations are also presented. In the final subsection, possible applications of the research are discussed.

5.1 Contributions to Research Objectives

In this section, the contributions outlined in Sec. 4.3 are related to the research objectives of the study. The relative contributions are presented in Tab. 5.1, and discussed in greater depth in the following subsections.

Table 5.1: Contributions to research objectives.

	RO1	RO2	RO3 (i)	RO3 (ii)	RO3 (iii)
RC1	++	++	+++	+	+
RC2	++	++	+++		
RC3	++	++			+++
RC4	++	++		+++	
RC5	++	++			++

RO1

Leverage machine learning to provide methods to support maritime situation awareness.

This thesis found that machine learning can be leveraged to support all levels of situation awareness. However, it was discovered that less focus in the research community has been on leveraging machine learning for level 3 situation awareness. The research in this thesis has, therefore, focused on methods that aid in providing level 3 situation awareness to ship navigators.

All the contributions presented in Sec. 4.3 consist of methods that leverage machine learning. As such, Tab. 5.1 shows that all research contributions address RO1.

RO2

Leverage historical AIS data to provide methods to support level 3 maritime situation awareness by artificially serving as navigational experience.

This thesis found that historical AIS data can be leveraged to support level 3 maritime situation awareness. The data was found to successfully serve as an artificial form of navigational experience when leveraged in conjunction with machine learning methods. AIS data were the basis for all the methods developed in the research papers. As a result, all contributions in Sec. 4.3 address RO2, as outlined in Tab. 5.1.

RO3

Develop methods to emulate the development of high level maritime situation awareness in humans by:

(i) *Developing methods to categorize ship behavior.*

To emulate human situation awareness, methods were developed to categorize ship behavior by first generating representations of historical AIS data, and subsequently clustering these representations. As such, RO3 (i) is addressed by RC1, which provides methods to generate representations of historical behavior, and RC2, which provides methods to cluster historical ship behavior.

(ii) *Developing methods to facilitate behavior models for predicting future ship behavior.*

In a similar manner to the development of human situation awareness, this thesis aimed to develop methods to predict future ship behavior through behavior models that relate to a specific category of ship behavior. RC4 addresses this objective by providing methods to predict future ship behavior. Furthermore, representations of the trajectories are leveraged to facilitate such predictions in some cases. As such, RC1 addresses this objective to some extent.

(iii) *Developing methods to facilitate pattern matching of observed ship behavior.*

To predict ship behavior, the relevant category must be identified. This is facilitated via pattern matching. RC3 directly addresses this objective by providing methods to classify a novel ship trajectory to a cluster of ship behavior. Furthermore, RC5 addresses this objective by providing methods to identify anomalous ship behavior. In this manner, a novel trajectory is identified as not belonging to any given pattern, i.e. category of ship behavior. RC1 also addresses this objective to some extent, as representations of AIS trajectories are leveraged to facilitate these functions.

5.2 General Discussion

In this section, a general discussion of the research outcome, in light of the research objectives, is presented.

5.2.1 Level 3 Situation Awareness Support

It is suggested that long-range trajectory predictions should support level 3 situation awareness. The results of this thesis indicate that by leveraging historical AIS data through machine learning, this can be achieved. In some cases, these predictions will likely exceed the capabilities of human navigators, especially those with limited experience.

Paper I showed that recent data-driven trajectory prediction techniques could be enhanced through decomposing the historical ship behavior into clusters of specific behavior. Paper II then improved upon these results with a novel trajectory

prediction approach that improved the accuracy of the predictions, as well as provided methods to estimate the uncertainty.

Deep learning techniques were also shown to be effective through a regional ship behavior prediction approach. These techniques also provide methods to decompose historical ship traffic into clusters of ship behavior, as well as predict future behavior accurately. If accurate, the developed methods in this thesis should aid navigators in conducting proactive collision avoidance measures, thereby enhancing the safety of maritime operations by avoiding close-range encounter situations.

Limitations

In cases where the automation system is incorrect, e.g. classification to an incorrect cluster of behavior, a situation awareness support system based on the methods in this thesis may have adverse effects. Such adverse effects may involve biasing effects as discussed in Sec. 2.1.2. The navigator must, therefore, constantly monitor the collision risk according to standard techniques, as in 2.3.1. The long-range predictions should only act as an outer layer that can prevent collision risk from arising. They should not replace classical techniques of collision avoidance once collision risk arises, triggering the applicability of the COLREGs.

Additionally, the methods developed in this thesis only present methods to facilitate trajectory predictions, and do not address how this should be presented to a navigator. As discussed in Sec. 2.1.2, situation awareness support, and not decision support, should be provided. As a result, it can be argued that alternative decisions should not be presented to the navigator. Endsley & Jones (2012) argued that the best way to support human performance is to support a high level of situation awareness. Hence, the estimated future trajectory and relevant parameters should be presented to the navigator.

Furthermore, the developed methods do not address how to handle cases in which both vessels in a two-vessel encounter situation apply a long-range trajectory prediction via the developed techniques. In such a case, both vessels may conduct proactive collision avoidance measures, rendering the trajectory predictions futile. This may, in turn, result in high risk situations that the algorithms are unable to foresee. Methods should, therefore, be developed to handle such situations.

5.2.2 Historical AIS Data

The findings in this thesis indicate that historical AIS data can be effectively utilized to infer the future trajectory of a selected vessel based on the past trajectories of similar vessels in the region. To facilitate this, machine learning methods were leveraged, where historical data were used to artificially represent navigational experience.

Historical AIS data from the region surrounding the city of Tromsø, Norway from the 1st of January, 2017 to the 1st of January 2018 were utilized. Using the AIS data set, there are innumerable test cases upon which the methods can be tested and validated against. The data in this region involve fairly complex ship traffic in inland waterways and around ports. As such, the region can be considered a good test case for the developed machine learning techniques, as opposed to regions characterized by more linear behavior.

The results indicate that leveraging AIS data facilitates long-range trajectory predictions that are not easily achieved via conventional techniques. Furthermore, the research indicates that AIS data can be leveraged to facilitate level 3 situation awareness projections with a level of accuracy useful for proactive collision avoidance. As such, it appears that AIS data is effective in artificially representing navigational experience for this purpose. The data were also conducive to the investigated deep learning approaches, due to the large size of the data set necessary for such techniques to be effective.

Limitations

The data set investigated in this thesis represents one year of historical AIS data. Ship behavior may change over time, and the methods should, therefore, be trained on updated data to best reflect the ship traffic in the region. Furthermore, the developed techniques have not been tested on other geographical regions. The models are also data-driven, and will not function in regions with no historical data. Regions with sparse data will likely also have degraded performance. Nonetheless, it can be argued that the investigated region includes a variety of complex behavior to evaluate the general performance of the methods.

Additionally, the data do not contain weather related parameters e.g. wind speed, wind direction, significant wave height, current direction, current speed and tidal data. Such parameters will likely have a significant impact on ship navigation, and should be included to enhance the performance of the prediction techniques. Many of the discovered clusters of ship behavior may in fact be due to various weather

conditions, and the inclusion of such data may, therefore, also aid in classification.

Furthermore, inherent errors in the data set, e.g. erroneous ship type, may degrade the methods. For instance, in Paper V, the ship type was used to aid in classification. If the ship types were considered to be discrete categories, such erroneous data could significantly degrade the performance of the classification. However, by embedding the ship types in Paper V, vector representations are learned, and used to evaluate the similarity between categories. It is likely that the learned embeddings of incorrectly labeled ships are close to the learned embeddings of the true labels in such cases, aiding the overall performance. Nonetheless, if entirely incorrect labels are applied (e.g. fishing vessel instead of cargo ship), the performance of the method will likely be degraded.

5.2.3 Machine Learning and Human Situation Awareness

As discussed in Sec. 2.1.1, it is suggested that humans develop mental models based on experience to facilitate situation awareness. It was argued that humans learn to categorize similar experiences together, and develop transition functions to model the future dynamics of the relevant system (i.e. behavior models). Furthermore, they are theorized to employ pattern matching to classify situations to one of these categories. In this thesis, it was found that machine learning could be applied to emulate the development of human situation awareness with respect to level 3 situation awareness. Machine learning techniques provide methods to:

- Generate meaningful representations of ship behavior.
- Cluster ship behavior (i.e. categorization).
- Predict future ship behavior (i.e. transition functions/behavior models).
- Classify observed ship behavior (i.e. pattern matching).

It was argued in Sec. 2.1.1 that humans generate abstract representations of situations, akin to the representations generated via dimensionality reduction and autoencoder approaches in machine learning. The results of this study have found that machine learning is effective in generating such representations. For the case of eigendecomposition-based dimensionality reduction, these representations are more easily interpreted. However, in the case of the investigated deep learning approaches, the representations are more abstract. The deep learning models learn to generate meaningful representations via multiple transformations of the data, that

can be more powerful than eigendecomposition-based techniques. Overall, the results indicate that machine learning a powerful tool to generate representations of historical ship behavior.

Clustering of historical behavior can also be considered analogous to human categorization of experiences. In this manner, the techniques employed to facilitate clustering of historical AIS trajectories emulate the manner in which humans categorize situations. In ship navigation, it is likely that navigators learn to categorize specific ship behavior based on experience. Furthermore, navigators likely classify the observed partial trajectory of a target vessel to one of the developed categories. Once a classification is conducted, the corresponding behavior model is employed to predict the future trajectory.

It was suggested in Sec. 2.1.1 that with more experience, operators develop a higher number of categories. In this sense, the behavior models in each category are able to model more specific behavior, resulting in more accurate predictions. As such, human development processes support the assertion in this thesis that generating prediction models with respect to specific behavior should enhance the accuracy.

However, the accuracy of such predictions will to a given extent rely on classification to the correct category and associated behavior model. In certain cases, however, the clusters (i.e. categories) may overlap, making classification challenging. Such categories will, however, likely have similar behavior models, resulting in good accuracy despite an incorrect classification. Humans likely apply such functions via pattern matching, and may also in certain cases match a situation to the incorrect schema. Similarly, if the schemata overlap to a certain extent, this may result in good accuracy despite incorrect classification. Nonetheless, classification accuracy should aid the performance of the predictions, where identifying specific behavior should enhance the fidelity of the predictions.

Limitations

Machine learning models are only as accurate as the data they are trained on. As such, if the behavior of a selected vessel does not match that of the historical behavior, the accuracy of the methods will likely be degraded. The models will also likely be biased towards the most commonly observed behavior.

Furthermore, tuning hyper-parameters in deep learning can be decisive for the performance. In this study, it is suggested to train deep learning models for specific regions of ship behavior. As such, the optimal hyper-parameters will vary.

5.2.4 Possible Applications

The methods developed in this thesis have been presented in the context of providing situation awareness to human navigators. However, the methods are also applicable for autonomous vessels. Given that autonomous vessels will require a form of situation awareness to facilitate collision avoidance, the techniques developed can be applied in the same way.

Autonomous vessels will rely on the ability to conduct effective collision avoidance maneuvers in close-range encounter situations. This entails that all systems must be operating at nominal levels, or a collision may occur. By preemptively predicting potential close-range encounter situations, such situations can be prevented by conducting proactive collision avoidance maneuvers. This also gives the autonomous vessel more time to conduct maneuvers before a critical situation arises, and enhances the overall safety of autonomous ship operations.

Furthermore, Vessel Traffic Service (VTS) centers can benefit from the developed methods. By estimating the future ship traffic in the region, they may be able to identify future hot spots that pose a higher risk of collision, and advise ships based on this. Close-encounter situations can also be predicted, and ships advised to preemptively avoid such situations.

Chapter 6

Concluding Remarks

This chapter presents the concluding remarks of the study. In the first section, the conclusions are outlined. Further work is then suggested in the next section.

6.1 Conclusions

The main objective of this study was to develop methods that can improve the safety of maritime transportation through enhanced maritime situation awareness. To facilitate this, it was suggested to leverage recent developments in machine learning and autonomous ship technology. Specifically, it was suggested to provide methods to aid level 3 situation awareness in ship navigation. Level 3 projections are likely used by ship navigators to simulate future ship traffic, and can be utilized for proactive collision avoidance measures. In this manner, future close-range encounter situations can be avoided, enhancing the overall safety of maritime operations. However, such actions must adhere to the COLREGs. As a result, developing methods to facilitate long-range trajectory predictions up to 30 minutes into the future were suggested to support such actions.

Inspired by the success of nature-inspired techniques, e.g. machine learning, it was suggested to emulate the manner in which humans develop high levels of situation awareness. Machine learning approaches were found to provide techniques that facilitate many of the same mechanisms, e.g. representation generation, clustering, pattern matching and prediction. However, maritime situation awareness is developed in humans through experience, where navigators likely predict future ship behavior based on the past behavior of similar vessels in the region. As a result, it was suggested to artificially represent navigational experience through historical ship behavior present in historical AIS data sets.

By leveraging machine learning and historical AIS data to emulate high level situation awareness in humans, this thesis has shown the feasibility of long-range ship behavior predictions. By decomposing historical ship behavior, categories of specific ship behavior can be discovered. The findings indicate that this facilitates accurate ship trajectory predictions. These predictions, if accurate, should support level 3 situation awareness in navigators.

Using eigenvector-based approaches, local trajectory segments can be extracted to represent the possible future behavior of a vessel for a specific prediction horizon. These methods can provide accurate predictions, and are relatively computationally efficient. As such, these techniques can be applied to facilitate live predictions. Furthermore, the methods can be applied to any region, as they extract the relevant data in an unsupervised manner, and do not require any input parameters.

Deep learning approaches have also proved to be powerful in generating meaningful representations of data. Via recurrent autoencoder approaches, it has been shown that deep learning can be utilized to discover the underlying structure of a set of historical AIS trajectories. Furthermore, by applying clustering algorithms to these representations, effective trajectory clustering can be facilitated. When applied to a cluster of ship behavior, deep learning was also found to be effective in predicting the future behavior of a selected vessel via a sequence-to-sequence model using attention. Once trained on a region of historical data, the developed deep learning framework can predict future ship behavior very efficiently.

Overall, this thesis has provided methods to support level 3 situation awareness. However, these methods must be implemented appropriately when supporting ship navigators. Future autonomous vessels may also benefit from the developed methods, as they will need to artificially achieve high level situation awareness. If successful, the developed methods can aid in improving the safety of future autonomous ship operations. Until autonomous ship operations are commonplace, however, autonomous ship technology will likely be used primarily as an aid to human navigators.

6.2 Suggestions for Further Work

While there has been much focus on utilizing AIS data for various tasks e.g. trajectory prediction in recent years, few studies focus on aiding proactive collision avoidance through level 3 situation awareness projections. As such, further work should support such objectives. The following extensions to the work are, therefore, suggested:

- Enhance the AIS data set by including relevant metocean parameters. By including such parameters, the predictive capabilities of the machine learning-based approaches should be enhanced.
- Develop methods to estimate the uncertainty of the predicted trajectories via deep learning approaches. One method may be to extend the work in Paper V by utilizing Monte Carlo Dropout as a Bayesian approximation (Gal & Ghahramani, 2016) in the output layer of the decoder.
- Investigate methods to facilitate automatic hyper-parameter tuning in the deep learning frameworks.
- Investigate alternative classification techniques in greater depth to improve classification accuracy.
- Investigate alternative prediction techniques in conjunction with the developed clustering and classification methods in greater depth.
- Research should be conducted on how to present such long-range trajectory predictions to navigators in an effective manner, including methods to evaluate the risk of future close-encounter situations based on such predictions.
- This thesis focuses on two-vessel encounter situations. Situations in which both vessels conduct proactive collision avoidance measures should be evaluated, as such situations may alter the future risk picture due to predictions not taking into account the proactive measures. Measures to ameliorate such issues should be addressed. Furthermore, multi-vessel encounter situations should be investigated, and how to leverage level 3 situation awareness projections effectively in such situations.
- Long-range prediction techniques should also be integrated with short-range predictions to facilitate more comprehensive predictor technology, as suggested in Perera & Murray (2019).
- This thesis suggests that long-range ship trajectory predictions should enhance the situation awareness of ship navigators based on relevant literature, but has not validated this assertion. As such, the situation awareness of navigators should be measured, both with and without a system that facilitates long-range trajectory predictions based on the techniques in this thesis. One approach may be via ship bridge simulators, where experienced and inexperienced navigators are invited to participate, e.g. as in Pedersen et al. (2003). Relevant measurement methods should be applied to compare their performance, as well as measure their situation awareness, such as those outlined in Endsley & Jones (2012).

Bibliography

- Bainbridge, L. (1983). Ironies of automation. In *Analysis, design and evaluation of man-machine systems* (pp. 129–135). Elsevier.
- Bartlett, F. (1932). *Remembering: A study in experimental and social psychology*. Cambridge University Press.
- Bellman, R. E. (1961). *Adaptive control processes: a guided tour* (Vol. 2045). Princeton university press.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. doi: 10.1109/72.279181
- Bloisi, D. D., Previtali, F., Pennisi, A., Nardi, D., & Fiorini, M. (2017). Enhancing automatic maritime surveillance systems with visual information. *IEEE Transactions on Intelligent Transportation Systems*, 18(4), 824-833. doi: 10.1109/TITS.2016.2591321
- Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5), 291–294. doi: 10.1007/BF00332918
- Braune, R., & Trollip, S. R. (1982). Towards an internal model in pilot training. *Aviation, space, and environmental medicine*, 53 10, 996-9.
- Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Advances in knowledge discovery and data mining* (pp. 160–172). Berlin, Heidelberg: Springer.
- Cane, T., & Ferryman, J. (2018). Evaluating deep semantic segmentation networks for object detection in maritime surveillance. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (p. 1-6). doi: 10.1109/AVSS.2018.8639077

- Carson-Jackson, J. (2012). Satellite ais – developing technology or existing capability? *Journal of Navigation*, 65(2), 303–321. doi: 10.1017/S037346331100066X
- Chan, C.-Y. (2017). Advancements, prospects, and impacts of automated driving systems. *International Journal of Transportation Science and Technology*, 6(3), 208–216. doi: 10.1016/J.IJTST.2017.07.008
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41.3(15). doi: 10.1145/1541880.1541882
- Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. Retrieved from <http://arxiv.org/abs/1409.1259>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *Nips'2014 deep learning workshop*. Retrieved from <http://arxiv.org/abs/1412.3555>
- Cockcroft, A. N., & Lameijer, J. N. F. (2011). *A guide to the collision avoidance rules* (7th ed.). Elsevier.
- Crocoll, W. M., & Coury, B. G. (1990). Status or Recommendation: Selecting the Type of Information for Decision Aiding. *Proceedings of the Human Factors Society Annual Meeting*, 34(19), 1524–1528. doi: 10.1177/154193129003401922
- 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)* (pp. 580–587).
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1–22.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).
- Endsley, M. R. (1988a). Design and evaluation for situation awareness enhancement. In *Proceedings of the human factors society annual meeting* (Vol. 32, pp. 97–101).
- Endsley, M. R. (1988b). Design and evaluation for situation awareness enhancement. *Proceedings of the Human Factors Society Annual Meeting*, 32(2), 97–101. doi: 10.1177/154193128803200221

- Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1), 32–64. doi: 10.1518/001872095779049543
- Endsley, M. R. (2017). From Here to Autonomy: Lessons Learned from Human-Automation Research. *Human Factors*, 59(1), 5–27. doi: 10.1177/0018720816681350
- Endsley, M. R., & Garland, D. J. (2000a). *Situation awareness analysis and measurement*. CRC Press.
- Endsley, M. R., & Garland, D. J. (2000b). Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, 1(1), 3–21.
- Endsley, M. R., & Jones, D. (2012). *Designing for situation awareness: An approach to user-centered design* (2nd ed.). CRC Press.
- Endsley, M. R., & Kris, E. O. (1994). Information presentation for expert systems in future fighters aircraft. *The International Journal of Aviation Psychology*, 4(4), 333-348. doi: 10.1207/s15327108ijap0404_3
- Eriksen, B. H., Wilthil, E. F., Flåten, A. L., Brekke, E. F., & Breivik, M. (2018). Radar-based maritime collision avoidance using dynamic window. In *2018 IEEE aerospace conference* (p. 1-9). doi: 10.1109/AERO.2018.8396666
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining* (p. 226–231). AAAI Press.
- Fabius, O., & van Amersfoort, J. R. (2015). Variational Recurrent Auto-Encoders. In *Proceedings of the international conference on learning representations (iclr)*. Retrieved from <http://arxiv.org/abs/1412.6581>
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 179–188. doi: 10.1111/j.1469-1809.1936.tb02137.x
- Forti, N., Millefiori, L. M., Braca, P., & Willett, P. (2020). Prediction of Vessel Trajectories from AIS Data Via Sequence-To-Sequence Recurrent Neural Networks. In *Icassp, IEEE international conference on acoustics, speech and signal processing - proceedings* (Vol. 2020-May, pp. 8936–8940). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICASSP40776.2020.9054421

- Fossen, T. I. (2000). A survey on nonlinear ship control: from theory to practice. *IFAC Proceedings Volumes*, 33(21), 1 - 16. (5th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2000), Aalborg, Denmark, 23-25 August 2000) doi: [https://doi.org/10.1016/S1474-6670\(17\)37044-1](https://doi.org/10.1016/S1474-6670(17)37044-1)
- Fujii, Y., & Tanaka, K. (1971). Traffic Capacity. *Journal of Navigation*, 24(4), 543–552. doi: 10.1017/S0373463300022384
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning (icml)* (pp. 1050–1059).
- Goerlandt, F., Montewka, J., Kuzmin, V., & Kujala, P. (2015). A risk-informed ship collision alert system: Framework and application. *Safety Science*, 77, 182–204. doi: 10.1016/j.ssci.2015.03.015
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Goodwin, E. M. (1975). A Statistical Study of Ship Domains. *Journal of Navigation*, 28(03), 328–344. doi: 10.1017/S0373463300041230
- Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Icassp, ieee international conference on acoustics, speech and signal processing - proceedings* (pp. 6645–6649). doi: 10.1109/ICASSP.2013.6638947
- Hancock, P. A. (2017). Imposing limits on autonomous systems. *Ergonomics*, 60(2), 284–291. doi: 10.1080/00140139.2016.1190035
- Harati-Mokhtari, A., Wall, A., Brooks, P., & Wang, J. (2007). Automatic identification system (AIS): Data reliability and human error implications. *Journal of Navigation*, 60(3), 373–389. doi: 10.1017/S0373463307004298
- Hermann, M., Pentek, T., & Otto, B. (2016). Design Principles for Industrie 4.0 Scenarios. In *2016 49th hawaii international conference on system sciences (hicss)* (pp. 3928–3937). IEEE. doi: 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. doi: 10.23919/ICIF.2017.8009762
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735
- Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA, USA: MIT Press.

- Huang, Y., Chen, L., Chen, P., Negenborn, R. R., & van Gelder, P. H. (2020). Ship collision avoidance methods: State-of-the-art. *Safety Science*, *121*, 451–473. doi: 10.1016/j.ssci.2019.09.018
- Huang, Y., van Gelder, P. H., & Wen, Y. (2018). Velocity obstacle algorithms for collision prevention at sea. *Ocean Engineering*, *151*, 308–321. doi: 10.1016/j.oceaneng.2018.01.001
- IMO. (2020). *Autonomous Shipping*. Retrieved from <https://www.imo.org/en/MediaCentre/HotTopics/Pages/Autonomous-shipping.aspx>
- Karhunen, K. (1946). Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae*, *37*.
- Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *Proceedings of the international conference on learning representations (iclr)*.
- Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. In *Proceedings of the international conference on learning representations (iclr)*. Retrieved from <http://arxiv.org/abs/1312.6114>
- Krogmann, U. (1999). From automation to autonomy: Trends towards autonomous combat systems. In *Advances in vehicle systems concepts and integration*. NATO Research and Technology Organization.
- Layton, C., Smith, P. J., & McCoy, C. E. (1994). Design of a cooperative problem-solving system for en-route flight planning: An empirical evaluation. *Human Factors*, *36*(1), 94-119. doi: 10.1177/001872089403600106
- Lee, E., Mokashi, A. J., Moon, S. Y., & Kim, G. (2019). The Maturity of Automatic Identification Systems (AIS) and Its Implications for Innovation. *Journal of Marine Science and Engineering*, *7*(9), 287. doi: 10.3390/jmse7090287
- Lefèvre, S., Vasquez, D., & Laugier, C. (2014). A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, *1*(1), 1–14.
- Lorenz, B., Di Nocera, F., Röttger, S., & Parasuraman, R. (2002). Automated fault-management in a simulated spaceflight micro-world. *Aviation, space, and environmental medicine*, *73*(9), 886—897.
- Madhavan, P., & Wiegmann, D. A. (2005). Cognitive anchoring on self-generated decisions reduces operator reliance on automated diagnostic aids. *Human Factors*, *47*(2), 332-341. doi: 10.1518/0018720054679489
- Mayer, R. (1983). *Thinking, problem solving, cognition*. W.H. Freeman.

- 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)* (pp. 1–6). IEEE. doi: 10.1109/SDF.2015.7347707
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133.
- Metzger, U., & Parasuraman, R. (2005). Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload. *Human Factors*, 47(1), 35–49. doi: 10.1518/0018720053653802
- Millefiori, L. M., Braca, P., Bryan, K., & Willett, P. (2016). Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction. *IEEE Transactions on Aerospace and Electronic Systems*, 52(5), 2313–2330. doi: 10.1109/TAES.2016.150596
- Murray, B., & Perera, L. P. (2018). A Data-Driven Approach to Vessel Trajectory Prediction for Safe Autonomous Ship Operations. In *Proceedings of the thirteenth international conference on digital information management (icdim)* (p. 240–247). IEEE. doi: 10.1109/ICDIM.2018.8847003
- Murray, B., & Perera, L. P. (2019). An AIS-based Multiple Trajectory Prediction Approach for Collision Avoidance in Future Vessels. In *Proceedings of the international conference on ocean, offshore and arctic engineering (omae)* (Vol. 7B-2019). American Society of Mechanical Engineers (ASME). doi: 10.1115/OMAE2019-95963
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International conference on machine learning (icml)*.
- Olson, W. A., & Sarter, N. B. (1999). Supporting informed consent in human-machine collaboration: The role of conflict type, time pressure, and display design. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 43(3), 189–193. doi: 10.1177/154193129904300313
- Onnasch, L., Wickens, C. D., Li, H., & Manzey, D. (2014). Human performance consequences of stages and levels of automation: An integrated meta-analysis. *Human Factors*, 56(3), 476–488. doi: 10.1177/0018720813501549
- 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. *Information Fusion (FUSION), 2014 17th International Conference on*(July), 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. doi: 10.3390/e15062218
- Pedersen, E., Inoue, K., & Tsugane, M. (2003). Simulator studies on a collision avoidance display that facilitates efficient and precise assessment of evasive manoeuvres in congested waterways. *Journal of Navigation*, *56*(3), 411–427. doi: 10.1017/S0373463303002388
- Perera, L. P. (2017a). Industrial iot to predictive analytics: A reverse engineering approach from shipping. In *Ceur workshop proceedings - 3rd norwegian big data symposium (nobids)*.
- Perera, L. P. (2017b). Navigation vector based ship maneuvering prediction. *Ocean Engineering*, *138*, 151–160. doi: 10.1016/j.oceaneng.2017.04.017
- Perera, L. P. (2020). Deep Learning Toward Autonomous Ship Navigation and Possible COLREGs Failures. *Journal of Offshore Mechanics and Arctic Engineering*, *142*(3). doi: 10.1115/1.4045372
- Perera, L. P., & Guedes Soares, C. (2015). Collision risk detection and quantification in ship navigation with integrated bridge systems. *Ocean Engineering*, *109*, 344–354. doi: 10.1016/j.oceaneng.2015.08.016
- Perera, L. P., & Murray, B. (2019). Situation Awareness of Autonomous Ship Navigation in a Mixed Environment Under Advanced Ship Predictor. In *Proceedings of the international conference on ocean, offshore and arctic engineering (omae)* (Vol. 7B-2019). American Society of Mechanical Engineers (ASME). doi: 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 Transactions on Intelligent Transportation Systems*, *13*(3), 1188–1200. doi: 10.1109/TITS.2012.2187282
- Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., & Quek, C. (2017). Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, *18*(8), 1993–2016. doi: 10.1109/TITS.2016.2634580
- Rasmussen, J., & Rouse, W. B. (1981). Human detection and diagnosis of system failures..
- Reichenbach, J., Onnasch, L., & Manzey, D. (2011). Human performance consequences of automated decision aids in states of sleep loss. *Human Factors*, *53*(6), 717–728. doi: 10.1177/0018720811418222

- Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital signal processing, 10*(1-3), 19–41.
- Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st international conference on international conference on machine learning - volume 32*.
- Ristic, B., Scala, B. L., Morelande, M., & Gordon, N. (2008). Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction. *2008 11th International Conference on Information Fusion*, 40–46. doi: 10.1109/ICIF.2008.4632190
- Riveiro, M., Pallotta, G., & Vespe, M. (2018). Maritime anomaly detection: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(5). doi: 10.1002/widm.1266
- 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 Engineering, 182*, 499–511. doi: 10.1016/J.OCEANENG.2019.04.024
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review, 65*(6), 386.
- Rothblum, A. M. (2000). Human error and marine safety. In *National safety council congress and expo, orlando, fl*.
- Rouse, W. B., & Morris, N. M. (1985). *On Looking Into the Black Box. Prospects and Limits in the Search for Mental Models* (Vol. 100) (No. 3). US: American Psychological Association. doi: 10.1037/0033-2909.100.3.349
- Rovira, E., McGarry, K., & Parasuraman, R. (2007). Effects of imperfect automation on decision making in a simulated command and control task. *Human Factors, 49*(1), 76-87. doi: 10.1518/001872007779598082
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. doi: 10.1038/323533a0
- Sarter, N. B., & Schroeder, B. (2001). Supporting decision making and action selection under time pressure and uncertainty: The case of in-flight icing. *Human Factors, 43*(4), 573-583. doi: 10.1518/001872001775870403

- Sarter, N. B., & Woods, D. D. (1991). Situation Awareness: A Critical But Ill-Defined Phenomenon. *The International Journal of Aviation Psychology*, 1(1), 45–57. doi: 10.1207/s15327108ijap0101_4
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. doi: 10.1109/78.650093
- Schwarz, G., et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2), 461–464.
- Sharma, A., Nazir, S., & Ernsten, J. (2019). Situation awareness information requirements for maritime navigation: A goal directed task analysis. *Safety Science*, 120, 745–752. doi: 10.1016/j.ssci.2019.08.016
- Spinner, T., Körner, J., Görtler, J., & Deussen, O. (2018). Towards an interpretable latent space : an intuitive comparison of autoencoders with variational autoencoders. In *Proceedings of the workshop on visualization for ai explainability 2018 (visxai)*. Retrieved from <https://thilosspinner.com/towards-an-interpretable-latent-space/>
- Srivastava, N., Mansimov, E., & Salakhutdinov, R. (2015). Unsupervised Learning of Video Representations using LSTMs. *32nd International Conference on Machine Learning, ICML 2015, 1*, 843–852. Retrieved from <http://arxiv.org/abs/1502.04681>
- Strauch, B. (2017). Ironies of automation: Still unresolved after all these years. *IEEE Transactions on Human-Machine Systems*, 48(5), 419–433.
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*, 4(January), 3104–3112. Retrieved from <http://arxiv.org/abs/1409.3215>
- Tam, C., & Bucknall, R. (2010). Collision risk assessment for ships. *Journal of Marine Science and Technology*, 15(3), 257–270. doi: 10.1007/s00773-010-0089-7
- Tam, C., Bucknall, R., & Greig, A. (2009). Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. *Journal of Navigation*, 62(03), 455. doi: 10.1017/S0373463308005134
- Theodoridis, S., & Koutroubas, K. (2009). *Pattern recognition* (4th ed.). Amsterdam: Elsevier.

- Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schuberth, P., . . . others (2016). Speeding up semantic segmentation for autonomous driving. In *Mlits, nips workshop* (Vol. 2).
- 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 Transactions on Intelligent Transportation Systems*, 1–24. doi: 10.1109/TITS.2017.2724551
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience, 2018*.
- Wickens, C., Hollands, J. G., Banbury, S., & Parasuraman, R. (1984). Engineering psychology and human performance.
- Wu, C. J. (1983). On the onvergence properties of the em algorithm. *The Annals of Statistics*, 95–103.
- Yang, J., Xiao, Y., Fang, Z., Zhang, N., Wang, L., & Li, T. (2017). An object detection and tracking system for unmanned surface vehicles. In K. U. Stein & R. Schleijsen (Eds.), *Target and background signatures iii* (Vol. 10432, pp. 244 – 251). SPIE. doi: 10.1117/12.2278220
- Üney, M., Millefiori, L. M., & Braca, P. (2019). Data driven vessel trajectory forecasting using stochastic generative models. In *Icassp 2019 - 2019 ieee international conference on acoustics, speech and signal processing (icassp)* (p. 8459-8463). doi: 10.1109/ICASSP.2019.8683444

Part III

Appended Papers

Paper I

Ship Behavior Prediction via Trajectory Extraction-Based Clustering for Maritime Situation Awareness

Brian Murray and Lokukaluge Prasad Perera (2021)

Submitted and in First Revision for Publication in *Journal of Ocean Engineering and Science*.

Ship Behavior Prediction via Trajectory Extraction-Based Clustering for Maritime Situation Awareness

Brian Murray*, Lokukaluge Prasad Perera

UiT The Arctic University of Norway, Tromsø, Norway

ARTICLE INFO

Keywords:

Maritime Situation Awareness
Ship Navigation
Trajectory Prediction
Collision Avoidance
Machine Learning
Unsupervised Learning
AIS

ABSTRACT

This study presents a method in which historical AIS data are used to predict the future trajectory of a selected vessel. This is facilitated via a system intelligence-based approach. This system intelligence can subsequently be utilized to provide enhanced situation awareness to navigators and future autonomous ships, such that collisions can more effectively be avoided. By evaluating the historical behavior of ships in a given geographical region, the method applies machine learning techniques to extrapolate commonalities in relevant trajectory segments. These commonalities represent historical ship behavior modes that correspond to the possible future behavior of the selected vessel. Subsequently, the selected vessel is classified to a behavior mode, and a trajectory with respect to this mode is predicted. This is achieved via an initial clustering technique and subsequent trajectory extraction. The extracted trajectories are then compressed using the Karhunen-Loève transform, and clustered using a Gaussian Mixture Model. The approach in this study differs from others in that trajectories are not clustered for an entire region, but generated to only correspond to the duration of a desired prediction horizon. As such, the extracted trajectories provide a much better basis for clustering relevant historical ship behavior modes. A selected vessel can then be classified to one of these modes using its observed behavior. Trajectory predictions are facilitated using an enhanced subset of data that likely correspond to the future behavior of the selected vessel. The method yields promising results, with high classification accuracy and low prediction error. However, vessels with abnormal behavior degrade the results in some situations, and have also been discussed in this study.

Nomenclature

a	Arbitrary AIS Parameter Vector	x	UTM x-coordinate [m]
A	Set of AIS Data	\mathbf{x}	Reduced Feature Vector
c	Trajectory Class	\mathbf{X}	Set of Reduced Feature Vectors
C	Data Cluster	y	UTM y-coordinate [m]
d	Euclidean Distance	\mathbf{z}	Class Membership
e	Eigenvector	\mathbf{Z}	Spatial Data Matrix
E	Eigenvector Matrix	ΔL	Step Size [m]
f	Trajectory Feature Vector	Λ	Eigenvalue Matrix
I	Identity Matrix	μ	Mean Vector
J_3	Class Separability Criterion	π	Prior Distribution
k	Hyper-parameter for k NN classifier	Σ	Covariance Matrix
K_M	Number of Free Parameters in Mixture Model	θ	Rotation Angle [$^\circ$]
L	Number of Data Points in Selected Trajectory	Θ	Model Parameters
$LL(\cdot)$	Log-likelihood Function	χ	Course over Ground [$^\circ$]
M	Number of Models in Mixture Model	<i>Subscripts</i>	
N	Number of Trajectories	0	Initial State
p	Arbitrary Vessel Position	i	Sample Number
q	Selected Vessel Position	j	Class Number
r	Search Radius [m]	k	k^{th} State
R	Rotation Matrix	l	Number of Eigenvectors
s	Vessel State	m	Model Number in Gaussian Mixture
S_b	Between-class Scatter Matrix	δ	Maximum Offset
S_w	Within-class Scatter Matrix	<i>Superscripts</i>	
T	Elapsed Time [s]	$\hat{\cdot}$	Estimated Parameter / State
T_δ	Additional Time Period [s]	<i>Acronyms</i>	
T_p	Desired Prediction Time Horizon [s]	AIS	Automatic Identification System
v	Speed over Ground [m/s]	BIC	Bayesian Information Criterion
		EM	Expectation Maximization
		GMM	Gaussian Mixture Model
		KL	Karhunen-Loève
		LDA	Linear Discriminant Analysis

 brian.murray@uit.no (B. Murray)
ORCID(s):

1. Introduction

Technological advances are permeating almost every industry. Artificial intelligence, increased computational power and wireless communication capabilities have the potential to allow for disruptive innovations that can change business models drastically. Many argue that there is a digital revolution underway and are calling it Industry 4.0 (Hermann et al., 2016). If one looks to the automotive industry for instance, significant innovations related to autonomous cars are being developed at an exponential rate. Autonomous cars are already being tested in general traffic areas and there are claims that mass production could be possible by 2021 (Chan, 2017).

Similarly, it can be argued that shipping is currently on its way into a fourth technical revolution, Shipping 4.0 (Rødseth et al., 2015). The first revolution in shipping can be argued to be the transition from sail to steam in at the turn of the 19th century, the second from steam to diesel around 1910, and the third came with the introduction of automated systems made possible through the advent of computers around 1970. Like the car industry, the shipping industry is looking to autonomy as a possible disruptive element. The shipping industry has, however, historically been considered conservative, with innovations being implemented at a slower rate than in similar industries. As such, technologies associated with autonomous ships are not as developed as those for autonomous cars. Nonetheless, many companies are working on the development of autonomous ships. The first autonomous ships, e.g. Yara Birkeland, are planned to be launched in 2020 and fully autonomous by 2022 (Yara, 2019). It can be argued that if the required technologies are available, autonomous ships will be safer and more efficient than conventional vessels, and that because of this fact they should be adopted by the industry (Levander, 2017). For this to occur, however, autonomous ships must be proven to operate at a level of safety comparable to, or better than, conventional manned vessels.

1.1. Maritime Situation Awareness

For autonomous ships to be introduced into commercial shipping lanes, effective collision avoidance systems (Perera et al., 2015) must be in place to ensure that the autonomous operations have the required level of safety. Given that the vessels are unmanned, an autonomous ship must be able to make decisions based on its understanding of its surroundings, i.e. its own situation awareness. Situation awareness is defined as *"being aware of what is happening around you and understanding what that information means to you now and in the future"* (Endsley et al., 2003) and is separated into three levels (Endsley, 1995):

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

For an autonomous vessel, situation awareness will primarily entail obstacle detection and prediction of close-range

encounter situations. Other vessels are the most common obstacle an autonomous ship will encounter and are referred to as target vessels in an encounter situation. The autonomous vessel in this case is referred to as the own ship. Such situations will require collision avoidance maneuvers.

1.1.1. Perception of Elements in The Current Situation

To effectively conduct collision avoidance maneuvers with respect to target vessels, an own ship will need to be able to first detect the target vessel and evaluate relevant parameters such as its position, course over ground and speed over ground. This can be considered as the first level of situation awareness. An autonomous ship must, therefore, first define its current state, where all obstacles and their current states are known. In order to perceive the relevant obstacles, an autonomous ship must be able to observe them. Since there is no navigator on-board, collision avoidance technologies will rely heavily on the sensor suite available on-board the vessel, as they must in essence replace the eyes of the navigator. An advanced obstacle detection and tracking system which utilizes sensor fusion to enhance detection capabilities should be utilized. Relevant sensors will likely include radar and electro-optical sensors (Prasad et al., 2017). Some examples include RADAR, LIDAR, stereo cameras and infra-red cameras.

1.1.2. Comprehension of The Current Situation

Based on its current state, the own ship must be capable of evaluating the risk of collision. If the risk of collision is identified as high, the own ship must conduct a collision avoidance maneuver that adheres to the COLREGS as outlined in Perera et al. (2010). This corresponds to level two of Endsley's situation awareness, where the ship must now make sense of its current state, and the immediate implications it has for the safety of the operation. Fujii and Tanaka (1971) and Goodwin (1975) introduced the concept of the ship domain, where a safety region around a relevant vessel is introduced to indicate the collision risk. A thorough review of collision avoidance methods can be found in Tam et al. (2009). These methods are designed with respect to ships in close-range encounters, where the collision risk is high enough to require collision avoidance maneuvers.

1.1.3. Projection of The Future Status

Level three of situation awareness addresses the projection of the future state of the vessel. In a collision avoidance setting, this entails predicting both the future states of the own ship as well as the future states of target vessels. Previous studies relating to collision avoidance techniques entail predicting the future state of a target vessel via calculations using constant course and speed values. Based on this, collision risk parameters relating to the closest point of approach (CPA) such as the distance (DCPA) and time (TCPA) can be determined, and necessary collision avoidance maneuvers conducted on this basis.

Ships have a slow response time when control actions are sent to change the speed or course over ground. Cars for instance can make changes almost instantaneously, depend-

ing on their speed. The inertia forces of a ship are, however, much higher, and resultant collision avoidance maneuvers will take much longer to conduct. Therefore, it is desirable to predict the risk of collision as far as possible in advance. This entails predicting the future trajectories of both the own ship and target vessels accurately. Methods such as Perera et al. (2011) where a fuzzy logic based decision making system for collision avoidance was introduced, and Yang et al. (2019), where parallel trajectory planning was proposed for autonomous collision avoidance, can improve the ability of an autonomous vessel to make decisions. Additionally, work on more advanced prediction algorithms, e.g. Perera et al. (2012), where extended Kalman filters were utilized to estimate ship trajectories, can enhance the situation awareness of autonomous vessels to aid in effective collision avoidance. However, predictions under such methods are only useful up to rather short prediction horizons (order of seconds to minutes). These methods are useful in the case of a close-range encounter situation, when the own ship must make decisions based on input from the sensor system and plan an effective collision avoidance maneuver. This however entails that the own ship already is in imminent danger.

This study suggests an approach in which the trajectory of a target vessel is predicted far in advance, such that a close-range encounter situation is prevented from occurring in the first place. The idea is that with an enhanced level of situation awareness, an autonomous vessel can predict its own future states, as well as those for relevant target vessels, for a period up to 30 minutes into the future. Based on this level of situation awareness, intelligent decisions can be made to identify possible future close-range encounter situations and optimally implement simple predictive collision avoidance strategies. Examples of such strategies could include minor speed or course alterations such that the future trajectory of the own ship is altered. This is unfortunately not straight forward. It can be assumed that the majority of vessels will be manned in the foreseeable future. As such, the behavior of potential target vessels is highly unpredictable for an autonomous agent. Such a strategy therefore requires a system intelligence based approach to maritime situation awareness.

1.2. System Intelligence Based Ship Trajectory Prediction

Data from the Automatic Identification System (AIS) provide a powerful dataset upon which analytics can be conducted. Historical AIS data provide insight into historical ship behavior that can be used to gain insight into patterns in maritime traffic. A myriad of ship parameters are recorded in the stored ship trajectories, including positional data, speed over ground values, and course over ground values for various time instances. AIS data provide an ideal dataset upon which machine learning techniques can be applied to yield insight into patterns for subsequent use in maritime traffic analysis. Machine learning is a very powerful field, where insight can be extracted from data for a variety of purposes. Examples in the maritime field include Xu et al. (2020), where

an optimal truncated least square support vector was utilized to estimate parameters for nonlinear maneuvering models, and Shen et al. (2019) where deep reinforcement learning was used to facilitate automatic collision avoidance.

This study suggests to provide future vessels with a degree of system intelligence, facilitated by historical knowledge that is extrapolated via machine learning techniques from AIS data. Using the historical knowledge available, such system intelligence will provide predictions of vessel trajectories, allowing for subsequent collision risk assessment. The purpose is to enhance the safety of both future autonomous ship operations, but also as a level of decision support to conventional vessels. This section presents relevant related work, and subsequently introduces the contributions of this study.

1.2.1. Related Work

An increasing amount of research is being conducted on methods to utilize AIS data. Zhang et al. (2017) for instance analyzed AIS data to gain insight into the demand and spatial-temporal dynamics of ship traffic around ports. Additionally, Liu et al. (2019) used AIS data to evaluate regional collision risk and Wen et al. (2020) utilized AIS data to automatically generate ship routes. Tu et al. (2017) provides a comprehensive review of methods to exploit AIS data for maritime navigation. Most work in the field has previously focused on predicting vessel trajectory patterns and general traffic behavior e.g. Gunnar Aarsæther and Moan (2009). Identifying anomalous behavior based on general vessel patterns e.g. Laxhammar et al. (2009) has also been of focus. These methods are useful for general behavior analysis, but are of limited use with respect to aiding in collision avoidance.

Of most interest in a collision avoidance setting is the work done on utilizing AIS data to predict the future trajectory of a vessel. The idea is to infer the future trajectory of a vessel based on the historical behavior of vessels in the same region stored in the historical AIS data. Ristic et al. (2008) presents a method to predict the future motion of a vessel utilizing a particle filter approach, but the accuracy is limited for use in collision avoidance. Pallotta et al. (2013) presents the TREAD (Traffic Route Extraction and Anomaly Detection) methodology to cluster all trajectories in a defined region in an unsupervised manner, and subsequently classify a selected vessel to one of the clusters, each representing a traffic route for the purpose of anomaly detection. Pallotta et al. (2014) subsequently utilized the TREAD methodology to identify traffic routes, classify a vessel to a route, and predict the vessel position along this route using the Ornstein-Uhlenbeck stochastic process. The TREAD technique, however, clusters waypoints, entry points, and stationary points such that the data for the entire region is utilized to differentiate between the vessels. As such, there can be significant discrepancies between sub-paths for trajectories belonging to the same class. This is of limited importance for long-term predictions (order of hours), and the the method using the Ornstein-Uhlenbeck stochastic process is effective

in such cases. The method's mean and variance functions do not change over time, however, which can be considered a strict assumption for real applications. Short-term predictions (order 5-30 minutes) of high accuracy and resolution, however, are arguably of more interest for collision avoidance purposes. For such predictions, the method will not be as effective. Mazzarella et al. (2015) also presents a prediction method using a Bayesian network based algorithm with a particle filter for prediction horizons in the order of hours, but also has limited efficacy in short-term trajectory predictions relevant for collision avoidance purposes.

Hexeberg et al. (2017) presents an AIS-based approach to predict short-term vessel trajectories. The method utilizes a single point neighbor search method to predict a vessel trajectory based on the underlying AIS data. The method, however, is unable to handle branching, and Dalsnes et al. (2018) expands on this work to provide multiple predictions via a prediction tree where samples are drawn from close neighbors in the underlying data. Predictions in this manner allow for a probability estimate to be evaluated for the future position for a given time horizon, and is facilitated via Gaussian mixture models. As opposed to previous methods, these methods do not utilize clustering to identify traffic routes. All predictions are based on the AIS data in the neighborhoods of predicted states. As such, these methods do not take into consideration the relationship between data points. Future states are predicted iteratively from an initial state based on the AIS data in the neighborhood of a predicted position. This data however may include data points that have no relationship to the initial or previous predicted states, and as such will degrade the accuracy. Rong et al. (2019) also presents an approach using a Gaussian Process model where a probabilistic trajectory prediction method is outlined which, in addition to predicting the future positions of a vessel, also describes the uncertainty of the predicted position. The method, however, is only evaluated with highly regular ship routes and offers no method to identify multiple possible future routes the vessel may follow and classify it to one.

1.2.2. Contribution

In this study, a method to provide system intelligence to future autonomous ships is suggested for the purpose of enhanced situation awareness. The method is facilitated by exploiting historical AIS data via machine learning techniques to predict the future trajectory of a vessel based on its initial state. The method provides short-term trajectory predictions (order 5-30 minutes) that can provide a basis for collision risk assessments. In this manner, possible close-range encounter situations can be avoided and the overall safety associated with autonomous operations can be increased.

The method presented in this study bases itself on a similar structure to that of previous techniques, in that trajectories are first clustered, a selected vessel is classified to a given cluster of trajectories, and a subsequent trajectory prediction is determined. However, this method is designed to aid in short-term trajectory predictions. As such, an alterna-

tive approach is suggested where an initial clustering technique is utilized to extract a subset of data from a historical AIS dataset centered about the initial vessel state. This cluster contains AIS data that has a high degree of similarity to the initial state of the selected vessel. Using this initial cluster, all unique future, i.e. forward, trajectories starting are extracted from the cluster. The length of these is defined by the desired prediction time horizon. These trajectories represent all future paths of ships that had similar states to the initial state of the selected vessel. This dataset will therefore only contain data that are relevant with respect to the initial vessel state, and will retain the relationship between data points.

The exacted forward trajectories represent the possible future behavior of the selected vessel for a given prediction horizon. It is of interest in this study to identify all possible trajectory modes of the historical ship behavior, such that a high fidelity trajectory prediction can be conducted to support collision avoidance. Identifying such modes can be facilitated by clustering the forward trajectories. It is only of interest to differentiate between different possible modes for the duration of the desired prediction horizon. As such, clustering the extracted forward trajectories will provide a better basis for relevant route identification compared to other methods where entire trajectories for regions are considered.

A clustering technique is suggested based on all relevant data in each unique extracted forward trajectory. Dimensionality reduction via the Karhunen-Loève transform is first conducted in order to compress the trajectories, whilst retaining the most important information relevant for differentiating the ship behavior. Such dimensionality reduction is necessary to allow for effective clustering. Clustering is then facilitated via unsupervised Gaussian Mixture Modeling. A selected vessel is then classified to a cluster based on its past behavior. This is achieved backward trajectory extraction, and optimally generating features for class separation using Linear Discriminant Analysis. Finally, a trajectory prediction is conducted with respect to the trajectory data in the cluster of historical ship behavior.

The method has enhanced performance as it can discover the cluster of most similar ship behavior. This allows for predictions with a higher degree of fidelity than other methods with respect to collision avoidance. This is effective for challenging regions with more complex traffic, i.e. multiple possible routes with various speeds. The trajectory prediction also provides an increased level of accuracy given the relationship between data points in the underlying data. Similar methods use techniques that introduce time invariance, such as dynamic time warping. These result in effective clustering of trajectories of similar shapes for a given region, but can not capture the relationship between when various behaviors are observed. Furthermore, clustering ship trajectories for entire regions will yield different results than clustering extracted trajectories for the desired prediction horizon, as suggested in this study. The technique in this study will provide a much better basis for differentiating relevant historical ship behavior used to predict the future trajectory of

a selected vessel. The method can also be applied in any geographical region, where the algorithm only requires access to raw AIS data of sufficient density of the region of interest.

An initial version of this work was presented in Murray and Perera (2019). Furthermore, in Murray and Perera (2020), a Dual Linear Autoencoder approach was introduced to facilitate trajectory prediction, that utilized similar clustering and classification regimes to those in this study. The clustering and classification techniques utilized in Murray and Perera (2020) were, however, not the focus of the study, and, therefore, not addressed in detail. This study, therefore, can be considered a parallel study, where the methods introduced in Murray and Perera (2019) are expanded upon, and addressed in detail.

2. Methodology

This section covers the methodology utilized to facilitate trajectory predictions via the proposed system intelligence approach. Given the current state of a target vessel, hereafter referred to as a selected vessel, the method predicts its future trajectory for a specified time horizon. The architecture of the method can be split into three modules; the trajectory clustering module, the trajectory classification module and the trajectory prediction module.

The three modules operate sequentially, with each operating on the output of the previous. In the first module, i.e. the trajectory clustering module, the user inputs characteristics of a selected vessel to the algorithm. The module will then identify other vessels with similar characteristics in the historical AIS dataset via an initial clustering technique. A forward trajectory extraction operation is then conducted, where the future trajectories of all the vessels in the initial cluster are extracted. The extracted forward trajectories are then clustered to identify groupings within the data. Each resultant cluster represents a possible route, or trajectory mode, the selected vessel may follow in the future. Cluster labels for all the trajectories present in the dataset are assigned, and can be utilized for further data analysis.

The objective of the second module is to determine which of the possible future paths the selected vessel will follow. This is achieved by first conducting a backward trajectory extraction operation, with class labels corresponding to those discovered in the clustering module. Comparing the past trajectory of the selected vessel, i.e. the trajectory prior to point provided to the algorithm as input, to the extracted backward trajectories, one can classify the selected vessel to one of the clusters discovered in the first module. This classification provides the most likely future route, or mode, the selected vessel will follow.

In the third module, a prediction of the future trajectory of the selected vessel is conducted using a subset of data corresponding to all extracted forward trajectories the belong to the same class, i.e. the same cluster. This yields a trajectory prediction along the most likely future route. Each module is addressed in detail in this section.

2.1. Trajectory Clustering Module

Machine learning can be split into two groups, supervised and unsupervised learning. Supervised learning deals with techniques where class labels are available, and one wishing to train an algorithm to correctly classify an unseen data point to a given class. Unsupervised learning, however, deals with data where the class labels are unavailable. In such a case, it is desirable to discover underlying groupings, or clusters, in the data. Clustering is, therefore, a form of unsupervised learning. In this study, the class labels for the extracted trajectories are unavailable, requiring the use of unsupervised learning. As such, clustering is investigated to discover groupings, or clusters, of historical ship trajectories that represent behavior modes that a selected vessel may belong to. This section covers the methodology utilized to cluster the historical trajectories.

2.1.1. Initial Clustering

The input to the algorithm is the initial state of a selected vessel, and is defined in (1). This state can be thought of as the current state of a target vessel whose future trajectory is of interest to predict. The parameters can be acquired from on board sensors e.g. radar, or from external sources e.g. AIS.

$$\mathbf{s}_0 \rightarrow [x_0, y_0, \chi_0, v_0, T_0] \quad (1)$$

It is of interest to identify similar vessels in the historical AIS database, i.e. data points with a high degree of similarity to \mathbf{s}_0 . It can be argued that AIS data points similar to \mathbf{s}_0 will have a higher probability of having similar trajectories than dissimilar data points. In essence, it is assumed that ships that were in a similar geographical location, with a similar course and speed over ground, will likely have behaved in a similar manner. In essence, the trajectories of these vessels can be thought of as representing the distribution of the future behavior of the selected vessel. As such, it is assumed that these trajectories can be used to estimate the future behavior of the selected vessel. The discovery of such similar vessels is achieved via the initial clustering technique described in this section.

A matrix \mathbf{Z} can be defined as the subset of AIS data solely containing spatial data. The spatial data is converted from longitude and latitude values to UTM coordinates (x, y) prior to clustering. A rotational affine transformation can be defined to rotate $\mathbf{Z} = [x_z, y_z]$ by $\theta = \chi_0$ to $\mathbf{Z}' = [x_{z'}, y_{z'}]$. This transformation is defined in 2.

$$\mathbf{Z}' = \mathbf{R} \mathbf{Z}^T \quad (2)$$

Where $x_z \in \mathbb{R}$, $y_z \in \mathbb{R}$, $x_{z'} \in \mathbb{R}$, $y_{z'} \in \mathbb{R}$ and \mathbf{R} is the rotation matrix defined as:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (3)$$

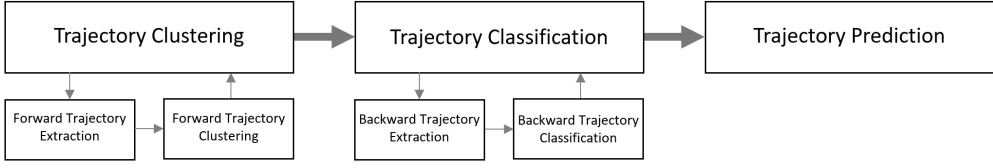
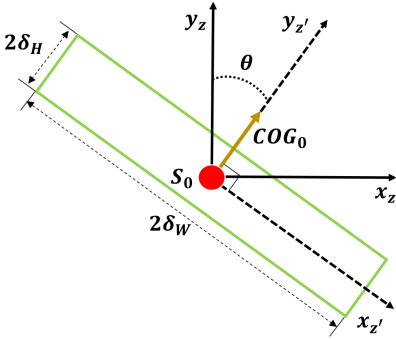


Figure 1: Method architecture


 Figure 2: Initial cluster C_0

The new matrix Z' will have a basis comprised of a vector in the direction of χ_0 and one orthogonal to χ_0 . An initial cluster C_0 is then created using data in the space spanned by these basis vectors in (4). This clustering operation results in a rectangular cluster C_0 with a height of $2\delta_H$ and width $2\delta_W$ centered about s_0 as illustrated in Fig. 2, which is adapted from that presented in Murray and Perera (2019). The cluster also only contains data points with similar χ and v values that were at a similar position to the selected vessel at some previous time point. The rectangular shape of the cluster orthogonal to χ_0 should capture most vessels that have similar trajectories to that of s_0 .

$$C_0 = \{\mathbf{a}_i \in \mathbf{A} : (|x_{z'i} - x_{z'0}| \leq \delta_W \wedge |y_{z'i} - y_{z'0}| \leq \delta_H) \wedge (|\chi_i - \chi_0| \leq \chi_\delta \wedge |v_i - v_0| \leq v_\delta)\} \quad (4)$$

2.1.2. Forward Trajectory Extraction

Based on the initial cluster C_0 , unique instances of vessel trajectories are identified, given that multiple data points in C_0 may belong to the same trajectory. Once unique trajectory instances have been identified, the nearest point of each trajectory to s_0 in geographical space is defined as its initial point. The forward trajectories of all instances are then extracted from this point and a period of time into the future corresponding to the desired prediction horizon T_p . An additional time period, T_δ , is extracted to ensure sufficient data density for the trajectory prediction module at the culmination of the prediction. The trajectories belonging to C_0 represent the possible behavior of the selected vessel, as their

initial points have a high degree of similarity to S_0 . In other words, it is likely that the future trajectory of the selected vessel will be similar to one of the trajectories in C_0 .

2.1.3. Trajectory Feature Generation

Assuming that the trajectories in C_0 represent the distribution of the possible future behavior of the selected vessel, it is desirable to discriminate between the various possibilities, i.e. discover groupings of behavior. In this sense, one wishes to cluster the trajectories into classes of behavior. To achieve this, each unique trajectory must be described by a set of features. The term feature in this case refers to an individual measurable parameter that describes the trajectory. Each trajectory is to be clustered in an unsupervised manner based on these features. As such, a trajectory feature vector is constructed comprising relevant parameters.

The first step in the generation of the feature vectors is to linearly interpolate each trajectory at 30 second intervals. This is done to generate higher density data, as well as provide a common time index with which the trajectories can be compared. The initial point of each trajectory is defined as T_0 , i.e. time zero. Subsequent data points are therefore 30 seconds apart, starting at this point. In this manner, the trajectories can be directly compared at the same time instance from T_0 . Using the interpolated data, each trajectory feature vector is constructed by flattening the matrix containing the positional and speed data (x, y, v) of the trajectory. If each trajectory is of length L , the resultant trajectory feature vector is defined as $\mathbf{f} \in \mathbb{R}^{3L \times 1}$. Utilizing the positional data, \mathbf{f} will incorporate the shape of the trajectory and the inherent course alterations between data points. The speed of the vessel along the trajectory will also be inherent in the positional data. Nonetheless, the speed over ground values at each time instance were deemed relevant to include to enhance the information stored in each vector.

As mentioned, the objective of the module is to cluster the trajectories, and as such the respective feature vectors should provide a basis for discriminating between the classes of behavior. Generally including as much information as possible, i.e. increasing the dimensionality of the feature vector, should enhance the discriminatory properties of the dataset. This is true, but in a clustering setting one runs into issues relating to the curse of dimensionality (Bellman, 1961).

Clustering is based on grouping data points via some distance measure. Points that are closer together are more likely to be considered part of the same cluster. The curse of di-

dimensionality in relation to clustering is discussed in Steinbach et al. (2004), where it is pointed out that a fixed number of data points will become increasingly sparse as the dimensionality increases. Data points can in a sense be lost in space as the dimensionality increases, as the distance between points with respect to a given dimension can be large. As a result, clustering data using standard techniques in a high dimensional space, will degrade the results, as the algorithms are unable to find groupings in the data. One method to ameliorate this effect is to reduce the dimensionality of the data.

A common method for dimensionality reduction is the the Karhunen-Loève (KL) transform (Karhunen, 1946). The purpose of the transform is to attain uncorrelated features and is shown in (5). First, the set of all feature vectors is centered such that all features have mean zero within the set. Subsequently, the covariance matrix Σ of the set of all feature vectors is calculated. Matrix \mathbf{E} consists of the eigenvectors of Σ , and Λ is the eigenvalue matrix, where the relationship is shown in (6). (5) projects the feature vector \mathbf{f} onto the space spanned by the eigenvectors of the covariance matrix. The covariance of the data inherently describes the correlation among the respective parameters. As such, the eigenvectors of the covariance matrix will describe the directions in which the data has the highest degree of variation orthogonal to each other.

$$\mathbf{x} = \mathbf{E}^T \mathbf{f} \quad (5)$$

Where $\mathbf{x} \in \mathbb{R}^{3L \times 1}$, $\mathbf{f} \in \mathbb{R}^{3L \times 1}$ and $\mathbf{E} \in \mathbb{R}^{3L \times 3L}$

$$\Sigma = \mathbf{E} \Lambda \mathbf{E}^T \quad (6)$$

Where $\Sigma \in \mathbb{R}^{3L \times 3L}$ and $\Lambda \in \mathbb{R}^{3L \times 3L}$

In a high dimensional space, however, many of the eigenvectors will describe very little variation in the data. The KL-transform, therefore, projects \mathbf{f} onto the subspace spanned by the l eigenvectors with the l largest eigenvalues in (7). This will inherently preserve the most important covariance information in the data whilst reducing the dimensionality to l . This may be abstract for the case of the trajectory feature vector, \mathbf{f} , as each dimension represents a positional or speed value at a given time instance. Take for instance the case of a 30 minute prediction with five minutes added to allow for sufficient data density. The dimensionality of \mathbf{f} will then be 210. The eigenvectors of Σ will point in the directions within this 210-dimensional space where there is a high degree of variation between the trajectories. As such, it is difficult to gain a direct physical interpretation of the eigenvectors, as the projection onto them represents a combination of multiple parameters. By choosing the l largest eigenvalues, one chooses the l directions where the variation in the data is greatest. When projecting the feature vectors onto the subspace spanned by the eigenvectors corresponding to the largest eigenvalues, one is in fact generating new

features with a high degree of variation that can be used for further analysis.

$$\mathbf{x} = \mathbf{E}_l^T \mathbf{f} \quad (7)$$

Where $\mathbf{x} \in \mathbb{R}^{l \times 1}$ and $\mathbf{E}_l \in \mathbb{R}^{3L \times l}$

In this study, the projection of \mathbf{f} onto the eigenvectors corresponding to the three largest eigenvalues was chosen as a representation for each trajectory. Generally, the projection should retain at least 95 % of the variance in the data. This is evaluated by investigating the sum of the chosen eigenvalues over the sum of all eigenvalues (Hyvärinen, 2009). It was found that using the the eigenvectors corresponding to the three largest eigenvalues fulfilled this requirement when evaluating the results. Additionally, a three-dimensional vector can easily be visualized when evaluating the performance of the clustering algorithm.

2.1.4. Unsupervised Gaussian Mixture Model Clustering

Using the reduced trajectory feature vectors generated via the KL-transform, the trajectories can be clustered. Depending on s_0 , the number of true clusters, i.e. classes, will vary. As such, a flexible clustering algorithm is required that can adapt to the data in each prediction. Unsupervised Gaussian Mixture Model Clustering was chosen for use in this study. A Gaussian Mixture Model (GMM) Reynolds et al. (2000) is a flexible model that adapts to the underlying data. GMMs assume that a set \mathbf{X} of data points consists of a mixture of M different Gaussian distributions. Each distribution has its own mean vector μ_m , covariance matrix Σ_m and prior distribution π_m . As such, each distribution will describe that particular class or cluster, i.e. class m . The class membership parameter, \mathbf{z}_i , is introduced for each data point \mathbf{x}_i where:

$$\mathbf{z}_{ik} = \begin{cases} 1 & \text{if } k = m \\ 0 & \text{otherwise} \end{cases}$$

Where $\mathbf{z}_i \in \mathbb{R}^{M \times 1}$

The class conditional probability is shown in (8). The most likely model is estimated by maximizing the log-likelihood with respect to the various model parameters.

$$p(\mathbf{x}_i | \mathbf{z}_{im} = 1) \sim N(\mu_m, \Sigma_m) \quad (8)$$

The class membership of the trajectories is, however, unknown. As such, the Expectation Maximization (EM) algorithm is utilized to conduct the unsupervised GMM clustering. The GMM requires that a specified number of underlying models M is input. Based on this, the EM algorithm initializes all model parameters. A common method is to initialize all μ_m as randomly chosen data points, the priors as $\pi_m = \frac{1}{M}$ and $\Sigma_m = \mathbf{I}$. This initialization is unlikely to model

the underlying data correctly. As such, the algorithm conducts what is known as the expectation step. In this step, the expected class membership $\langle \mathbf{z}_{im} \rangle$ is evaluated in (9), based on the current model parameters, Θ . All data points will, therefore, have updated class memberships based on the current model parameters.

$$\langle \mathbf{z}_{im} \rangle = \frac{p(\mathbf{x}_i | \mathbf{z}_{im} = 1; \Theta) \pi_m}{\sum_{k=1}^M p(\mathbf{x}_i | \mathbf{z}_{ik} = 1; \Theta) \pi_k} \quad (9)$$

The next step in the EM algorithm is known as the maximization step. In this step, the model parameters are updated based on the new distribution resulting from the expectation step. This is done by maximizing the log-likelihood with respect to Θ . The estimated parameters in the maximization step are calculated in (10), (11) and (12).

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{i=1}^N \langle \mathbf{z}_{im} \rangle \mathbf{x}_i}{\sum_{i=1}^N \langle \mathbf{z}_{im} \rangle} \quad (10)$$

$$\hat{\boldsymbol{\Sigma}}_m = \frac{\sum_{i=1}^N \langle \mathbf{z}_{im} \rangle (\mathbf{x}_i - \boldsymbol{\mu}_m)(\mathbf{x}_i - \boldsymbol{\mu}_m)^T}{\sum_{i=1}^N \langle \mathbf{z}_{im} \rangle} \quad (11)$$

$$\hat{\pi}_m = \frac{\sum_{i=1}^N \langle \mathbf{z}_{im} \rangle}{N} \quad (12)$$

The EM algorithm now repeats, where the expected class memberships are updated, and subsequently the model parameters. The algorithm is in a sense adapting to the data, where the most likely distribution of the data is discovered. This iterative process continues in a loop until a stopping criteria is met. One common stopping criteria is the convergence of the total log-likelihood. Alternatively, one can terminate the algorithm if there is little to no change in the model parameters, i.e. the parameters themselves converge. The parameter convergence criteria was utilized in this study. Often times, the EM algorithm can have issues with convergence, due to poor initialization. To avoid divergence issues, a technique often known as "N random starts" was utilized, where N different initializations are run for a number of iterations. The best run, i.e. the run with the greatest log-likelihood score, is then chosen and run for further iterations. The mixture model will, upon convergence, consist of M distinct Gaussian distributions which describe the class conditional probabilities, $p(\mathbf{x} | c_m)$, of the data, along with an associated prior distribution, π_m . The posterior probability $p(c_m | \mathbf{x})$ can be found via Bayes Rule in (13) using the resultant conditional probabilities and priors from the algorithm.

$$p(c_m | \mathbf{x}) = \frac{p(\mathbf{x} | c_m) \pi_m}{p(\mathbf{x})} \quad (13)$$

$$p(c_m | \mathbf{x}) > p(c_j | \mathbf{x}) \forall j \neq m, j = 1 \dots M \quad (14)$$

Clustering of the dataset is then conducted via Bayesian classification, where each feature, \mathbf{x}_i , is classified to class m according to (14). However, the number of underlying classes M is as previously mentioned unknown. In order to determine the most likely number of clusters, the Bayesian Information Criterion (BIC) (Schwarz et al., 1978) defined in (15), is utilized.

$$BIC = -2LL(\Theta_M) + K_M \ln(N) \quad (15)$$

For a GMM with M underlying distributions, $LL(\Theta_M)$ is the total log-likelihood function computed at the optimum, K_M the number of free parameters in the mixture model, and N the number of data points. The EM algorithm can be run for various GMMs by altering M . By calculating the BIC for each resultant GMM, the most likely GMM is that with the lowest BIC. This is due to it having the highest likelihood and least complexity. In this study, it was assumed that there will be no more than 20 unique clusters in the trajectory data, and the BIC was, therefore, evaluated for values of M up to 20.

This process discovers the best GMM to fit the data and provides the number of possible routes, or trajectory behavior modes, a selected vessel may belong to. By classifying all the extracted forward trajectories, class labels can assigned. These labels are used for further analysis in the subsequent modules.

2.2. Trajectory Classification Module

The trajectory clustering module has now clustered all trajectories present in \mathbf{C}_0 to M classes. Each class represents a group of trajectories that have a high degree of similarity. As such, each class represents a possible future route, or behavior mode, the selected vessel may belong to. It is now of interest to classify the selected vessel to the most likely class of the M possibilities. In this sense, an estimate of the distribution of the possible future behavior of the selected vessel can be made. Using the data in the class of trajectory behavior, a trajectory prediction can be made. This section presents the method utilized to achieve such a classification.

2.2.1. Backward Trajectory Extraction

One possible method to conduct the aforementioned classification is to utilize the current vessel state, S_0 , and compare them to the data points in \mathbf{C}_0 . This, however, will have limited predictive power, as the classification will be based solely on one time instance of the selected vessel. An alternative approach is, therefore, suggested, where the previous 10 minutes of the selected vessel's trajectory are compared to the previous 10 minutes of data for all trajectories in \mathbf{C}_0 . This in a sense is the inverse of the forward trajectory extraction process described in Sec. 2.1.3. Instead of extracting the trajectories from T_0 and for instance 30 minutes into the future, the past trajectories are extracted from the same initial point, i.e. from T_0 , and 10 minutes into the past from that time instance. 10 minutes was chosen, as it assumed that at least 10 minutes of behavior for the selected

vessel should be available via the on board sensors of the own ship, or via external sources e.g. AIS. The method is otherwise identical to that described in Sec. 2.1.3. All the backward trajectories extracted from \mathbf{C}_0 will have the same labels as those determined by the clustering technique in Sec. 2.1.4. As such, a labeled dataset is available that can be used to classify the observed trajectory of selected vessel.

2.2.2. Optimal Feature Generation

Each backward trajectory feature is represented by the flattening of the matrix containing all position and speed over ground data in the same manner as the forward trajectories in Sec. 2.1.3. This will result in a vector $\mathbf{f} \in \mathbb{R}^{3L \times 1}$. In the case of a 10 minute trajectory this will be a 60-dimensional space within which the classification must take place. This can be a challenging task, as it is likely that the features are quite similar, given that the vessels in \mathbf{C}_0 generally will have similar trajectories for the past 10 minutes.

To improve the classification accuracy, Linear Discriminant Analysis (LDA) (Fischer, 1936) is utilized. LDA provides a method to generate features with optimal separation between classes in a supervised manner. Using the class separability measure J_3 in (16), one can optimize a transformation such that features are generated to optimize class separability.

$$J_3 = \text{trace}\{\mathbf{S}_w^{-1}\mathbf{S}_m\} \quad (16)$$

\mathbf{S}_m is the mixture scatter matrix defined as $\mathbf{S}_m = \mathbf{S}_w + \mathbf{S}_b$, where \mathbf{S}_w is the within-class scatter matrix and \mathbf{S}_b the between-class scatter matrix. \mathbf{S}_w and \mathbf{S}_b are defined in (17) and (19) respectively. \mathbf{S}_w describes how compact the data within each class is, whilst \mathbf{S}_b describes how spread out each class is with respect to the global mean. In a classification setting, one wishes to minimize the trace of \mathbf{S}_w , i.e. data are more compact within each class, and maximize the trace of \mathbf{S}_b , i.e. the classes are more spread out. This corresponds to maximizing the class separation criterion J_3 .

$$\mathbf{S}_w = \sum_{m=1}^M \pi_m \mathbf{\Sigma}_m \quad (17)$$

$$\boldsymbol{\mu}_0 = \sum_{m=1}^M \pi_m \boldsymbol{\mu}_m \quad (18)$$

$$\mathbf{S}_b = \sum_{m=1}^M \pi_m (\boldsymbol{\mu}_m - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_m - \boldsymbol{\mu}_0)^T \quad (19)$$

It is desirable to find a transformation $\mathbf{x} = \mathbf{A}^T \mathbf{f}$ such that J_3 is maximized in the transformed space. The optimal transformation with respect to class separability is found to be $\mathbf{A} = \mathbf{E}$ where \mathbf{E} is the matrix of eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$

in the original vector space. This relationship is shown in (21) where $\boldsymbol{\Lambda}$ is the corresponding diagonal eigenvalue matrix. The transformation is shown in (20). However, \mathbf{S}_b is of rank $M - 1$, and correspondingly $\mathbf{S}_w^{-1}\mathbf{S}_b$ is also of rank $M - 1$. As such, there will be $M - 1$ nonzero eigenvalues. (20) will, therefore, project \mathbf{f} onto the subspace spanned by the l largest eigenvectors in a similar manner to the KL-transform. If $l = M - 1$, optimality with respect to J_3 will be preserved. Further dimensionality reduction can still be conducted by choosing a value $l < M - 1$. This will, however, be a suboptimal solution. Further details on LDA can be found in Theodoridis and Koutroumbas (2009).

$$\mathbf{x} = \mathbf{E}^T \mathbf{f} \quad (20)$$

Where $\mathbf{x} \in \mathbb{R}^{3L \times 1}$, $\mathbf{f} \in \mathbb{R}^{3L \times 1}$ and $\mathbf{E} \in \mathbb{R}^{3L \times l}$

$$\mathbf{S}_w^{-1}\mathbf{S}_b = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T \quad (21)$$

Where $\mathbf{S}_w^{-1}\mathbf{S}_b \in \mathbb{R}^{3L \times 3L}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{l \times l}$

2.2.3. Classification

Despite utilizing the optimal features described in Sec. 2.2.2, the classification task is highly non-linear, and likely with significant overlap between classes in most cases. This is due to the high degree of similarity between the past trajectories. As a result, the k -Nearest Neighbor (k NN) classifier (Dasarathy, 1991) is utilized due to its nonlinear predictive power.

Given a data point \mathbf{x}_0 , the k NN classifier will measure the distance to all other data points, \mathbf{x}_i , in the dataset \mathbf{X} using the Euclidean distance as shown in (22).

$$d_i = \|\mathbf{x}_i - \mathbf{x}_0\|_2 \quad (22)$$

The k NN classifier will then identify the k nearest data points using distance measures from (22). Based on this subset of data, the algorithm then identifies the class with the most data points in the subset and classifies \mathbf{x}_0 to the majority class.

In this study, \mathbf{x}_0 is the projection of the backward trajectory feature vector \mathbf{f}_0 of the selected vessel onto the LDA subspace according to (20). The k NN classification is then conducted in the LDA subspace, where the k nearest trajectories, i.e. most similar, are found and the majority class is defined as the class, j , of the selected vessel. The corresponding subset for class j , $\mathbf{A}_j \subset \mathbf{A}$, represents the set of AIS data points that belong to the trajectories in this class.

2.3. Trajectory Prediction Module

Once the past trajectory of the selected vessel has been classified to one of the classes, a trajectory prediction can be conducted with respect to that class. The input data to this module are all trajectory data belonging to the class determined in Sec. 2.2.3, i.e. \mathbf{A}_j . This approach assumes that

the classification is accurate, and as such only predicts one unique trajectory.

The algorithm utilized for the trajectory prediction is adapted from that presented in Murray and Perera (2018). The algorithm is based on the method outlined in Hexeberg et al. (2017) where a Single Point Neighbor Search Method was presented to predict vessel trajectories based on historical AIS data.

Given the initial state of the selected vessel S_0 , the prediction algorithm estimates the future states of the selected vessel. This is an iterative process where the state in the k^{th} iteration is defined in (23).

$$\hat{\mathbf{s}}_k \rightarrow [\hat{x}_k, \hat{y}_k, \hat{\chi}_k, \hat{v}_k, \hat{T}_k] \quad (23)$$

The estimated future position in state k , i.e. $[\hat{x}_k, \hat{y}_k]$, is estimated given the parameters in state $\hat{\mathbf{s}}_{k-1}$ as a distance Δ_L from $[\hat{x}_{k-1}, \hat{y}_{k-1}]$ in the direction of $\hat{\chi}_{k-1}$. The time parameter \hat{T}_k is then updated according to (24).

$$\hat{T}_k = \hat{T}_{k-1} + \frac{\Delta L}{\hat{v}_{k-1}} \quad (24)$$

Once the position parameters $[\hat{x}_k, \hat{y}_k]$ are updated, $\hat{\chi}_k$ and \hat{v}_k are updated using a circular distance based clustering technique. A cluster \mathbf{C}_k can be defined according to (25) where \mathbf{p}_i is an arbitrary vessel position, and \mathbf{q}_k is the selected vessel position at $\hat{\mathbf{s}}_k$. The clustering is conducted on the set of data points in the subset of AIS data that corresponds to the classified class, i.e. \mathbf{A}_j . \mathbf{C}_k will, therefore, be defined as the points within a radius r from the predicted position.

$$\mathbf{C}_k = \{\mathbf{a}_i \in \mathbf{A}_j : \|\mathbf{p}_i - \mathbf{q}_k\| \leq r\} \quad (25)$$

$\hat{\chi}_k$ and \hat{v}_k are defined as the median values of the points in cluster \mathbf{C}_k according to (26) and (27). The median values were chosen as opposed to the mean given that they are less sensitive to outliers.

$$\hat{\chi}_k = \text{median}(\chi_i \in \mathbf{C}_k) \quad (26)$$

$$\hat{v}_k = \text{median}(v_i \in \mathbf{C}_k) \quad (27)$$

This iterative process continues until the desired prediction horizon is reached, i.e. $\hat{T}_k \geq T_p$. The set of all estimated states will constitute the predicted trajectory of the selected vessel. The predicted trajectory is subsequently linearly interpolated at 30 second intervals for comparative analysis. The method is illustrated in Fig. 3 adapted from that presented in Murray and Perera (2019).

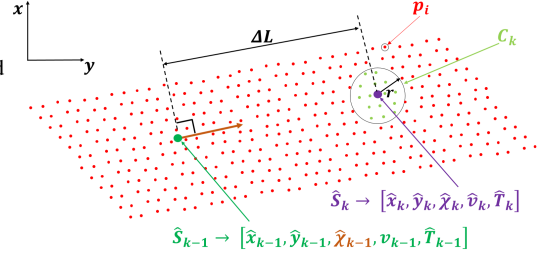


Figure 3: Illustration of trajectory prediction technique

3. Results and Discussion

This section presents results from a case study. The future trajectory of a selected vessel was predicted using the outlined method in this study. 100 data points were randomly selected from a dataset corresponding to one year of AIS data from January 1st 2017 to January 1st 2018 for the region around the city of Tromsø, Norway. This dataset corresponded to approximately 15 million AIS data points. Each data point was initialized as the initial state for the 100 selected vessels. The case study predicted the future 30 minute trajectory for each selected vessel, where the predictions were validated using the true trajectories stored in the historical AIS data. The performance of each module was investigated, as well as the overall performance of the method in predicting future ship trajectories.

3.1. Trajectory Clustering Module

3.1.1. Extracted Trajectories

As outlined in Sec. 2.1.2, all trajectories present in the initial cluster \mathbf{C}_0 centered about the initial vessel state, \mathbf{s}_0 were extracted. An example of the interpolated extracted trajectories is visualized in Fig. 4. The illustrated position data are defined with respect to \mathbf{s}_0 (i.e. $[x_0, y_0] = [0, 0]$) to more easily visualize the distances involved. To the human eye, it is evident that there are two main routes the vessel may follow, with a few outliers. This information may also be what a navigator on the bridge might be aware of, and base his future decisions upon.

3.1.2. Clustering Results

The first phase of the clustering technique is to reduce the dimensionality according to (7). Subsequently, a GMM is fit to the projection of the trajectory data in the subspace spanned by the three eigenvectors with the largest eigenvalues, as outlined in Sec. 2.1.4. This technique was found to be quite effective in generating new features with a high degree of variation between data points. The result of the clustering of the extracted forward trajectories in Fig. 4 is visualized in Fig. 5. This figure illustrates the clusters in the reduced subspace. Using the discovered classes, the labeled trajectories are visualized in Fig. 6. The method in this example has discovered eight unique clusters. This implies that the vessel may have one of eight behavior modes. It is evident that the

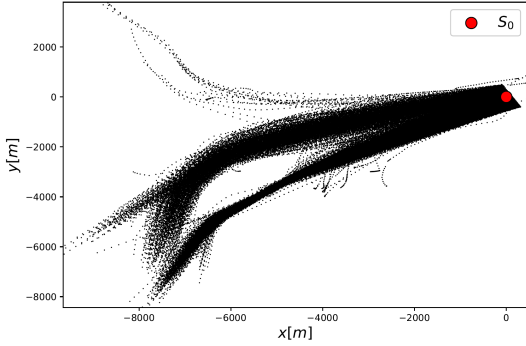


Figure 4: Illustration of extracted forward trajectories

algorithm has primarily focused on differences in the spatial aspects of the trajectories, i.e. the upper and lower routes. However, the results indicate that the algorithm also discovers sub-routes within the main routes. These indicate vessels traveling along the prevailing route at various speeds. As such, the algorithm is in fact discovering behavior modes within the data. The method, therefore, is effective in regions with more complex traffic. In regions where vessels have a high degree of regularity, such advanced clustering will likely not be as necessary.

The results indicate that KL-transform effectively stores the most important information from the trajectories by projecting a 210-dimensional vector to a 3-dimensional vector. This data compression subsequently allows for effective clustering in the lower-dimensional subspace, where multiple trajectory groupings can be discovered, providing a more accurate dataset upon which a trajectory prediction can be conducted. Given that the true future trajectory of the selected vessel is also available in the historical data, it can be classified to one of the clusters via the GMM. This provides the true class of the selected vessel for subsequent accuracy analysis.

3.2. Trajectory Classification Module

3.2.1. Optimal Feature Representation

In this phase of the method, the backward trajectories of all vessels present in the initial cluster, C_0 , are extracted. These trajectories are visualized for the example in Fig. 7 with labels from the corresponding forward trajectories. The motivation now is to classify the past trajectory of the selected vessel to one of the classes. Using (20), the trajectory features are projected onto the LDA-subspace. In this subspace, the trajectories are optimally separated, making it easier for classification. The projection onto the three largest components for the previous example is visualized in Fig. 8.

3.2.2. Classification

Using a k NN classifier with $k = 7$, the projection of the backward trajectory of the selected vessel was classified to one of the clusters. This resulted in the selected vessel being classified to the purple class previously illustrated. The

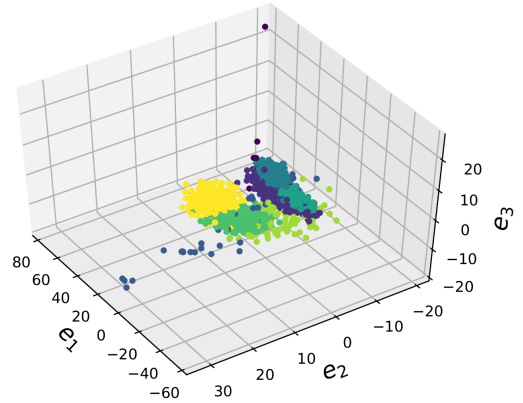


Figure 5: Clustering results in KL-subspace

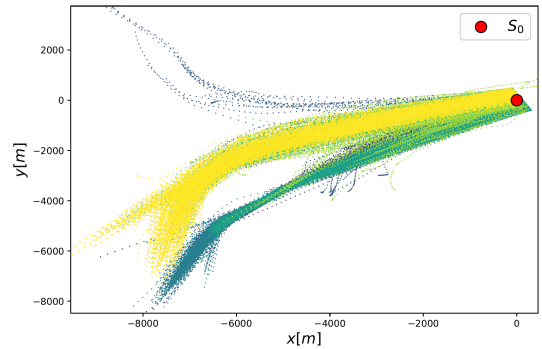


Figure 6: Labeled forward trajectories

extracted forward trajectory data corresponding to this class is illustrated in Fig. 9. The value of k was iteratively varied, and proved to have little effect on the classification accuracy. As a result, a value of $k = 7$ was utilized.

3.3. Trajectory prediction module

Using the data visualized in Fig. 9, a prediction can be conducted utilizing the methodology outlined in Sec. 2.3. Fig. 10 illustrates the resultant prediction for the previous example. For this case, the prediction appears to closely correspond to the true vessel trajectory.

3.4. Prediction Accuracy

In order to evaluate the overall performance of the method, 100 random data points were chosen from the AIS dataset. Each data point was defined as the initial state of a selected vessel, i.e. s_0 . The method outlined in Sec. 2 was then run on each selected vessel to predict its future trajectory. In this section, the accuracy of the classification module and the position error of the resultant trajectory predictions are evaluated.

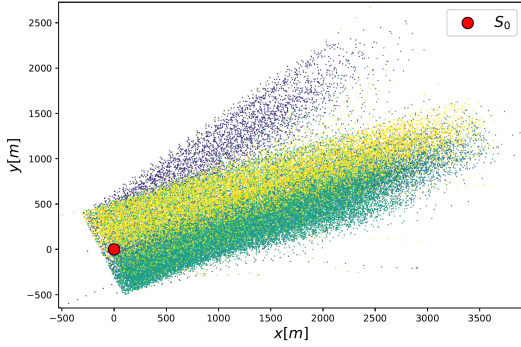


Figure 7: Backward trajectories with labels from the corresponding forward trajectories

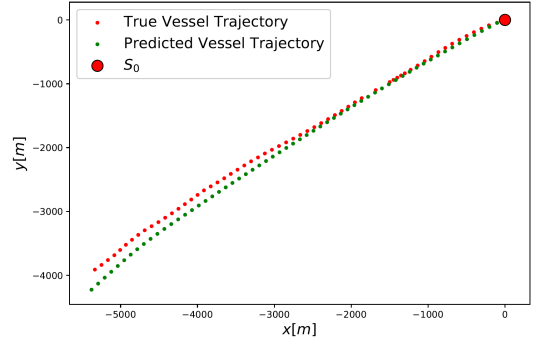


Figure 10: Trajectory prediction

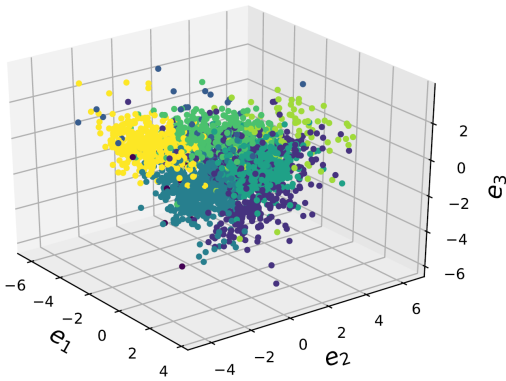


Figure 8: LDA projection of backward trajectory data

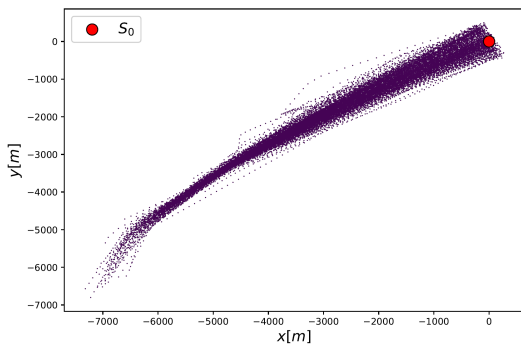


Figure 9: Forward trajectories corresponding to class prediction

3.4.1. Classification Accuracy

The true class label of the selected vessel, i.e. the ground truth, was evaluated using the fitted GMM for each tested vessel. The predicted classes for all vessels were then compared with the ground truth, and an overall classification accuracy calculated. It was found that for the 100 cases tested

in this study, the classification accuracy was of 70 %. This indicates that the features generated via LDA from the backward trajectories provided a basis to correctly classify 70 of the 100 tested vessels.

3.4.2. Position Accuracy

The position accuracy of the trajectory predictions was also investigated. The accuracy was evaluated as a function of time, where the distance between the true and predicted position of the selected vessel define the error. The position error was calculated for three cases; the overall error for all vessels, the error for incorrectly classified vessels, and the error for correctly classified vessels. Given that the true trajectories of the selected vessels are of various lengths, the position error is evaluated as a percentage of the true distance traveled for each time instance. The distance traveled for each selected vessel was estimated as the sum of trajectory segments extracted from the true trajectory.

The median position errors for all cases are illustrated in Fig. 11. The median error was chosen for presentation as opposed to the root mean squared error due to the sensitivity of the root mean squared error to outliers. It is clear that the error is significantly higher for the incorrectly classified vessels. However, for those vessels which are classified to the correct class, the median error is quite reasonable with a value of approximately 4 % of the true distance traveled for a 30 minute prediction. The error appears to increase rather linearly. This is to be expected, as errors will accumulate as a function of time.

The position error of the incorrectly classified vessels is also investigated, as the method mis-classifies 30 % of the vessels, and as such, will have a corresponding performance in these cases. Fig. 12 illustrates an example of a trajectory prediction when the selected vessel was incorrectly classified. It is evident that the subsequent error can grow to be quite high for the case of a 30 minute prediction. For the case of the incorrectly classified vessels in Fig. 11, a sudden dip is observed around a prediction horizon of 15 minutes. This effect is not observed for the correctly classified vessels. It is likely due to the nonlinear nature of many of the ship trajectories. At this point, certain ships that are predicted to travel

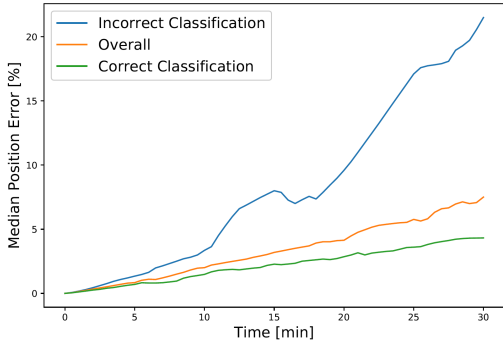


Figure 11: Median position error of trajectory predictions evaluated as a percent of the distance traveled

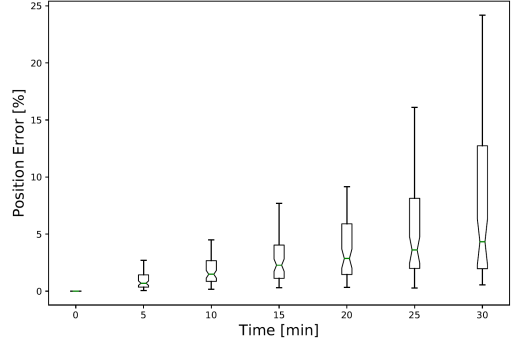


Figure 13: Prediction error for correctly classified vessels evaluated as a percent of the distance traveled

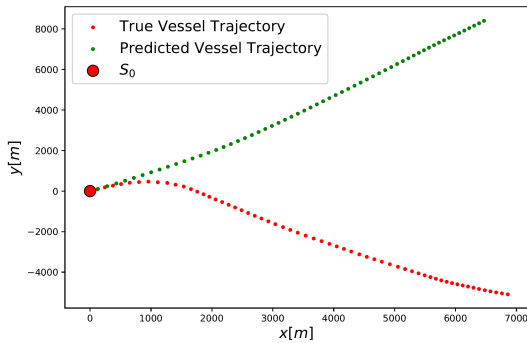


Figure 12: Incorrectly classified trajectory prediction

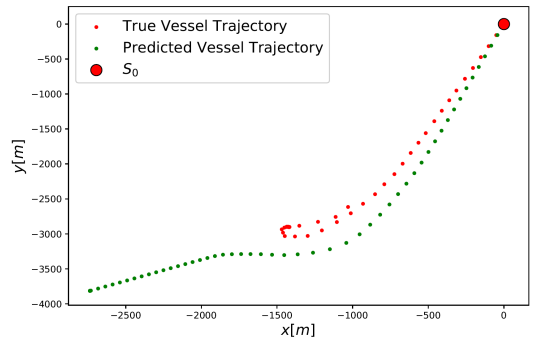


Figure 14: Correctly classified trajectory with high position error

along an incorrect route, turn and approach the true route, causing the position error to decrease for a short period of time, before once again linearly increasing.

In order to reduce the error associated with the incorrect classifications, one needs to improve the classification module to provide either a better representation to conduct classification on, or utilize another classifier. Additionally, anomalies can be filtered out as they will have a degrading effect on the classification. The error with respect to the correctly classified predictions is therefore of greater interest for further investigation. Fig. 13 illustrates the box plots for the position error at five minute intervals. The horizontal green line illustrates the median error. It appears that the lower 50 % of the predictions are rather tightly bounded, whilst the upper 50 % have a higher variance. The variance of the error also increases significantly as a function of time. This is to be expected, as the predictions are dependent on both the speed estimates, as well as the degree of variation within the cluster.

Nonetheless, the degree of variance observed for the 30 minute predictions is quite high for the upper quartiles. The correctly classified vessels with poor predictions were therefore investigated. Fig. 14 visualizes one such case. It appears that the vessel has been classified correctly, as the pre-

dicted and true trajectories are similar at first. However, approximately half way along the predicted trajectory, the true trajectory of the vessel stops and turns around. Such irregular ship trajectories are difficult to predict, and the algorithm is unable to identify and recreate such patterns. These irregularities are the source of much of the high positional error illustrated in Fig. 13. The algorithm is, nonetheless, effective in predicting what can be considered regular ship trajectories, represented by the lower quartiles in Fig. 13. In general, it appears that the approach yields successful results with respect to predicting the future trajectory of a selected vessel.

4. Conclusion and Further Work

This study presents a method to provide system intelligence to future autonomous ships such that they achieve a level of maritime situation awareness. This is facilitated through the use of historical AIS data and machine learning. Relevant trajectories are extracted from historical AIS data, and commonalities in the data are discovered via Gaussian Mixture Model clustering. These clusters represent modes of historical ship behavior. When predicting the future trajectory of a target vessel, it is likely that its future behavior

will belong to one of these modes. Therefore, the observed behavior of a selected vessel is classified to one of the modes to improve the fidelity of a trajectory prediction. Such high fidelity predictions can then be used to aid in collision avoidance.

Assuming that the previous behavior of the selected vessel is known, the method has a high classification accuracy. The results indicate that the use of Linear Discriminant Analysis provides a more optimal basis for classification. However, if the previous behavior is unknown, the classification accuracy will likely be degraded.

The results for the trajectory predictions indicate that the method was able to successfully predict the future trajectory of a selected vessel, with relatively low error. For misclassified vessels, however, the performance was significantly degraded. This is to be expected, given that predictions are conducted with respect to incorrect behavior modes in these cases.

For the cases investigated in this study, correctly classified vessels had low prediction error for time horizons up to 30 minutes. The median error value was approximately 4 % of the true distance traveled after 30 minutes. This is likely aided by the direct relationship between data points in the clusters utilized in the predictions, as well as the ability to discover ship behavior modes that match the selected vessel. Certain vessels, however, had anomalous behavior, which the method was unable to accurately predict.

The method presented in this study is generic, and can be applied to any geographical region, given sufficient density of the historical AIS data. As a result, the algorithm can be implemented in a generic form on any vessel, and will run on the raw AIS data for that region. Seeing as the method is data-driven, the amount of data available will enhance the results. The accuracy of the predictions will also be location specific, as the number of possible behavior modes that exist will vary. The method will likely have better performance in open waterways with fewer possible routes, and a generally high degree of regularity in ship behavior compared to more complex waterways such as coastal regions and ports. The sensitivity of location has, however, not been investigated. Nonetheless, the ability of the method to discover behavior modes in the historical data will improve the performance in complex waterways compared to other methods. The predictions are also conducted without considering the prevailing weather conditions. These will likely have a significant effect of the behavior of a vessel, and should be included in the prediction method.

Further work will include enhancing the classification accuracy of the method, as well as including weather parameters into the dataset. Considering that the classifier utilized was a k NN classifier, the accuracy can likely be further increased by using more advanced architectures e.g. support vector machines (Vapnik, 1995). Alternative trajectory prediction methodologies applying additional machine learning techniques, e.g. deep learning, will also be investigated to further enhance the predictions. It is also vital to connect the trajectory predictions to existing collision avoidance frame-

works and regulations. This will be addressed in the future, where such predictions will be applied in a collision avoidance setting.

Acknowledgements

An initial version of this paper was presented at the 38th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2019), June, 2019, Glasgow, Scotland. 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, Industry and Fisheries. The authors would also like to express gratitude to the Norwegian Coastal Administration for providing access to their AIS database.

References

- Bellman, R.E., 1961. Adaptive control processes: a guided tour. volume 2045. Princeton university press.
- Chan, C.Y., 2017. Advancements, prospects, and impacts of automated driving systems. *International Journal of Transportation Science and Technology* 6, 208–216. doi:10.1016/j.ijst.2017.07.008.
- 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.
- Dasarathy, B.V., 1991. Nearest neighbor (NN) norms : NN pattern classification techniques. IEEE Computer Society Press, Los Alamitos, California.
- Endsley, M.R., 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37, 32–64. doi:10.1518/001872095779049543.
- Endsley, M.R., Bolte, B., Jones, D.G., 2003. Designing for situation awareness : an approach to user-centered design. Taylor & Francis.
- Fischer, R.A., 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x.
- Fujii, Y., Tanaka, K., 1971. Traffic Capacity. *Journal of Navigation* 24, 543–552. doi:10.1017/S0373463300022384.
- Goodwin, E.M., 1975. A Statistical Study of Ship Domains. *Journal of Navigation* 28, 328–344. doi:10.1017/S0373463300041230.
- Gunnar Aarsæther, K., Moan, T., 2009. Estimating Navigation Patterns from AIS. *Journal of Navigation* 62, 587–607. doi:10.1017/S0373463309990129.
- 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. doi:10.1109/HICSS.2016.488.
- Hexeberg, S., Flåten, 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. doi:10.23919/ICIF.2017.8009762.
- Hyvärinen, A., 2009. Principal component analysis. URL: http://www.mv.helsinki.fi/home/amoaning/movies/uml/pca_ghandout.pdf.
- Karhunen, K., 1946. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae* 37.
- Laxhammar, R., Falkman, G., Sviestins, E., 2009. Anomaly detection in sea traffic—a comparison of the gaussian mixture model and the kernel density estimator, in: Information Fusion, 2009. FUSION'09. 12th International Conference on, IEEE. pp. 756–763.
- Levander, O., 2017. Autonomous ships on the high seas. *IEEE Spectrum* 54, 26–31.
- Liu, Z., Wu, Z., Zheng, Z., 2019. A novel framework for regional collision

- risk identification based on AIS data. *Applied Ocean Research* 89, 261–272. doi:10.1016/j.apor.2019.05.020.
- 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. doi:10.1109/SDF.2015.7347707.
- Murray, B., Perera, L.P., 2018. A Data-Driven Approach to Vessel Trajectory Prediction for Safe Autonomous Ship Operations, in: Proceedings of The Thirteenth International Conference on Digital Information Management (ICDIM), Berlin., pp. 240–247. doi:10.1109/ICDIM.2018.8847003.
- 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 (OMAE 2019), Glasgow, Scotland. doi:10.1115/OMAE2019-95963.
- Murray, B., Perera, L.P., 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., 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. *Information Fusion (FUSION)*, 2014 17th International Conference on , 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, 2218–2245. URL: <http://www.mdpi.com/1099-4300/15/6/2218>, doi:10.3390/e15062218.
- Perera, L.P., Carvalho, J.P., Guedes Soares, C., 2010. Autonomous guidance and navigation based on the COLREGs rules and regulations of collision avoidance. *Advanced Ship Design for Pollution Prevention* , 205–216doi:10.1201/b10565-26.
- Perera, L.P., Carvalho, J.P., Guedes Soares, C., 2011. Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. *Journal of Marine Science and Technology* 16, 84–99. doi:10.1007/s00773-010-0106-x.
- Perera, L.P., Ferrari, V., Santos, F.P., Hinostrroza, M.A., Guedes Soares, C., 2015. Experimental Evaluations on Ship Autonomous Navigation and Collision Avoidance by Intelligent Guidance. *IEEE Journal of Oceanic Engineering* 40, 374–387. doi:10.1109/JOE.2014.2304793.
- Perera, L.P., Oliveira, P., Guedes Soares, C., 2012. Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction. *IEEE Transactions on Intelligent Transportation Systems* 13, 1188–1200. doi:10.1109/TITS.2012.2187282.
- Prasad, D.K., Rajan, D., Rachmawati, L., Rajabally, E., Quek, C., 2017. Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment: A Survey. *IEEE Transactions on Intelligent Transportation Systems* 18, 1993–2016. doi:10.1109/TITS.2016.2634580.
- Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted gaussian mixture models. *Digital signal processing* 10, 19–41.
- Ristic, B., Scala, B.L., Morelande, M., Gordon, N., 2008. Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction. 2008 11th International Conference on Information Fusion , 40–46doi: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 Engineering* 182, 499–511. doi:10.1016/J.OCEANENG.2019.04.024.
- Schwarz, G., et al., 1978. Estimating the dimension of a model. *The annals of statistics* 6, 461–464.
- Shen, H., Hashimoto, H., Matsuda, A., Taniguchi, Y., Terada, D., Guo, C., 2019. Automatic collision avoidance of multiple ships based on deep Q-learning. *Applied Ocean Research* 86, 268–288. doi:10.1016/j.apor.2019.02.020.
- Steinbach, M., Ertöz, L., Kumar, V., 2004. The Challenges of Clustering High Dimensional Data, in: *New Directions in Statistical Physics*. Springer Berlin Heidelberg, 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. *Journal of Navigation* 62, 455–476. doi:10.1017/S0373463308005134.
- Theodoridis, S., Koutroumbas, K., 2009. *Pattern recognition*. 4th ed. ed., Elsevier, Amsterdam.
- 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 Transactions on Intelligent Transportation Systems* , 1–24doi:10.1109/TITS.2017.2724551.
- Vapnik, V.N., 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.
- Wen, Y., Sui, Z., Zhou, C., Xiao, C., Chen, Q., Han, D., Zhang, Y., 2020. Automatic ship route design between two ports: A data-driven method. *Applied Ocean Research* 96, 102049. doi:10.1016/j.apor.2019.102049.
- Xu, H., Hassani, V., Guedes Soares, C., 2020. Comparing generic and vectorial nonlinear manoeuvring models and parameter estimation using optimal truncated least square support vector machine. *Applied Ocean Research* 97, 102061. doi:10.1016/j.apor.2020.102061.
- Yang, R., Xu, J., Wang, X., Zhou, Q., 2019. Parallel trajectory planning for shipborne Autonomous collision avoidance system. *Applied Ocean Research* 91, 101875. doi:10.1016/j.apor.2019.101875.
- Yara, 2019. Yara Birkeland press kit. URL: <https://www.yara.com/news-and-media/press-kits/yara-birkeland-press-kit/>.
- Zhang, L., Meng, Q., Fang Fwa, T., 2017. Big AIS data based spatial-temporal analyses of ship traffic in Singapore port waters. *Transportation Research Part E: Logistics and Transportation Review* doi:10.1016/J.TRE.2017.07.011.

Paper II

A Dual Linear Autoencoder Approach for Vessel Trajectory Prediction using Historical AIS data

Brian Murray and Lokukaluge Prasad Perera (2020)

Published in *Ocean Engineering*, 209, 107478.

<https://doi.org/10.1016/j.oceaneng.2020.107478>



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).

Nomenclature

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

Subscripts

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

Superscripts

^	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. [Aarsaether 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, way-points 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

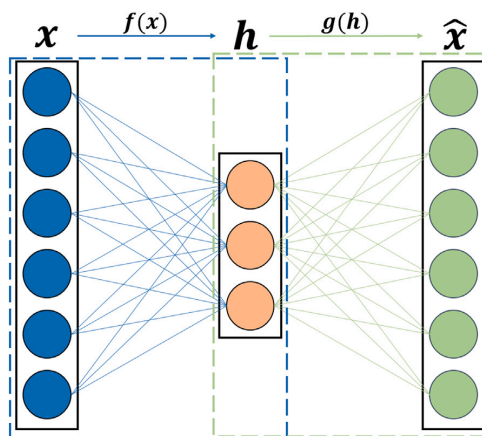


Fig. 1. Autoencoder architecture with the encoder illustrated by the blue dashed box, and the decoder by the green dashed box. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 $f(x)$ that produces the code, h , shown in (1), and a decoder function, $g(h)$, that reconstructs the data from the code shown in (2). An integral part of an autoencoder is the internal hidden layer, h , that represents the code space, often referred to as the latent representation of the data.

$$h = f(x) \quad (1)$$

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

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 h has a smaller dimension than 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, $f(x)$, 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.

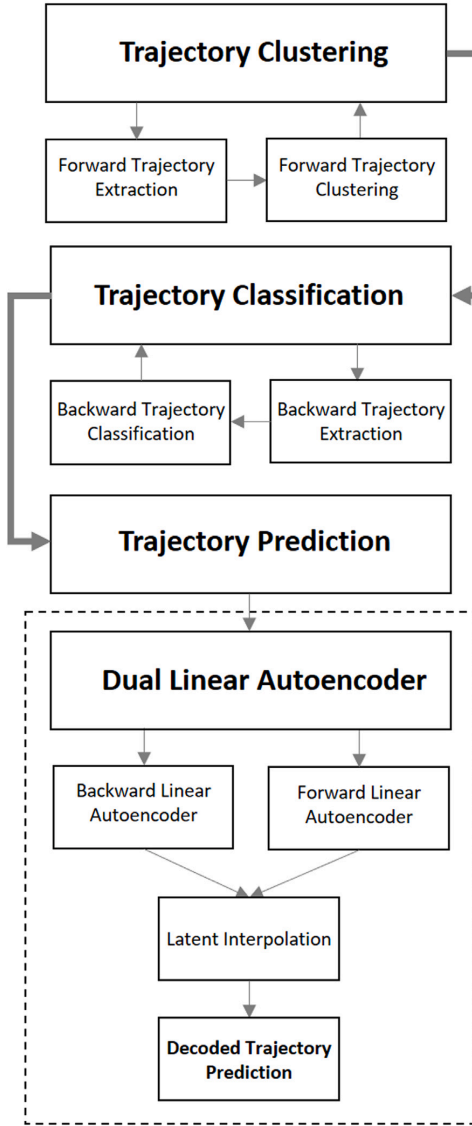


Fig. 2. Prediction architecture, with the dual linear autoencoder block within the dashed lines.

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.

$$s_0 \rightarrow [x_0, y_0, \chi_0, v_0] \quad (3)$$

The method first identifies historical AIS data points with a high degree of similarity to 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 C_0 is created. 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. C_0 is defined according to the following equation (Murray and Perera, 2019):

$$C_0 = \{a_i \in A : (|x_{z'_i} - x_{z'_0}| \leq \delta_W \wedge |y_{z'_i} - y_{z'_0}| \leq \delta_H) \wedge (|\chi_i - \chi_0| \leq \chi_\delta \wedge |v_i - v_0| \leq v_\delta)\} \quad (4)$$

Additionally, data points that do not match the ship type of the selected vessel are removed. 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 s_0 in each unique trajectory determined. C_0 is then updated by filtering out all data points other than these most similar points. In this manner, 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 C_0 , the forward trajectory from the corresponding point in C_0 is extracted. This can be thought of as the future trajectory defined in relation to the point in 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 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

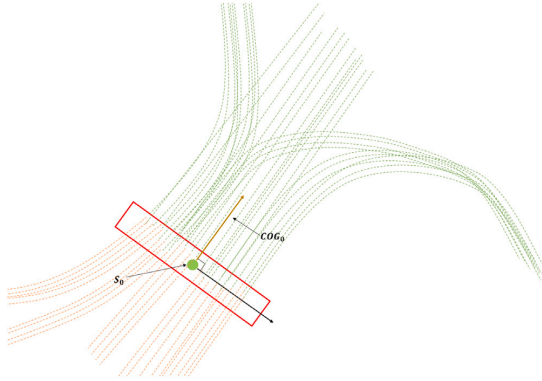


Fig. 3. C_0 illustrated as a rectangular red box. The forward and backward extracted trajectories illustrated as green and orange lines respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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–Loève (KL) transform (Karhunen, 1946) in (5), where the dimensionality is reduced from $2L$ to l .

$$\mathbf{h}_f = \mathbf{E}^T \mathbf{f} \quad (5)$$

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

$$\Sigma = \mathbf{E} \mathbf{A} \mathbf{E}^T \quad (6)$$

where $\Sigma \in \mathbb{R}^{2L \times 2L}$ and $\mathbf{A} \in \mathbb{R}^{2L \times 2L}$

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 μ_m , covariance matrix Σ_m and prior distribution π_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 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 k NN classifier is utilized.

$$\mathbf{h}_b = \mathbf{E}^T \mathbf{b} \quad (7)$$

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

$$\mathbf{S}_{wc}^{-1} \mathbf{S}_{bc} = \mathbf{E} \mathbf{A} \mathbf{E}^T \quad (8)$$

where $\mathbf{S}_{wc}^{-1} \mathbf{S}_{bc} \in \mathbb{R}^{2L \times 2L}$ and $\mathbf{A} \in \mathbb{R}^{l \times l}$.

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 Σ 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.

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

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

$$J = E [\|\mathbf{x} - \hat{\mathbf{x}}\|^2] \quad (11)$$

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.

$$\mathbf{h}_f = \mathbf{E}_f^T \mathbf{f} \quad (12)$$

$$\hat{\mathbf{f}} = \mathbf{E}_f \mathbf{h}_f \quad (13)$$

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.

$$\mathbf{h}_b = \mathbf{E}_b^T \mathbf{b} \quad (14)$$

$$\hat{\mathbf{b}} = \mathbf{E}_b \mathbf{h}_b \quad (15)$$

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.

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

$$w_i = \frac{1}{d_i} \quad (17)$$

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

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, s_j , where each state is separated by 30 s.

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

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.

$$\mathbf{H}_{f,s} \sim \mathcal{N}(\hat{\mathbf{h}}_{f,s}, \Sigma_{h,s}) \quad (20)$$

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

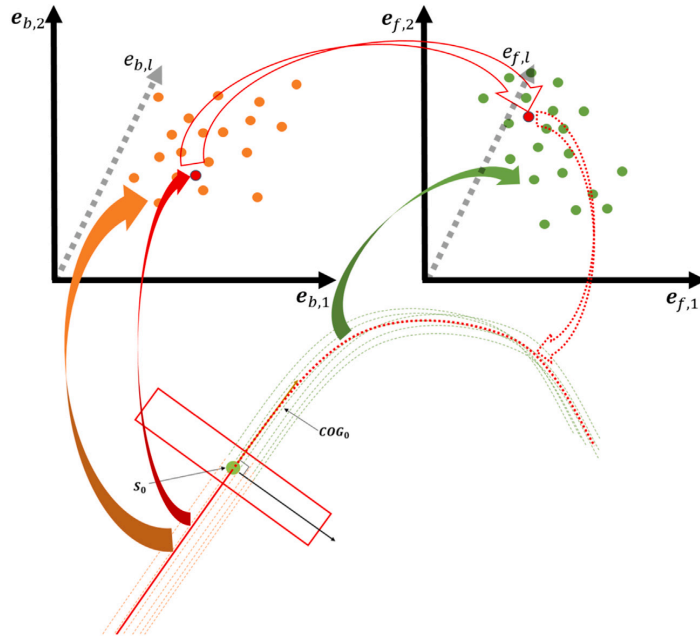


Fig. 4. Illustration of the latent representations in the dual linear autoencoder architecture. The interpolation between the known latent representation of the backward trajectory of the selected vessel to the unknown forward latent representation is also illustrated. (For interpretation of the references 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 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.

$$\mathbf{P}_j \sim \mathcal{N}(\mu_j, \Sigma_j) \tag{22}$$

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, 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 k NN 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

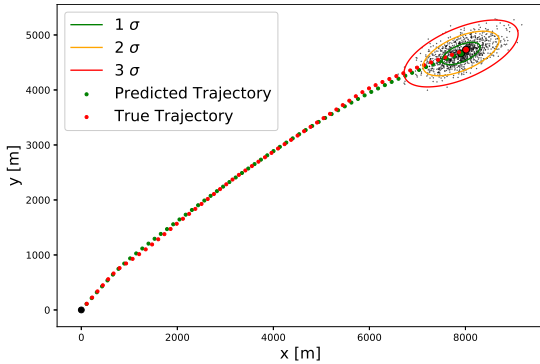


Fig. 5. Trajectory prediction with correct classification. Contours are illustrated for $T_p = 30$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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.

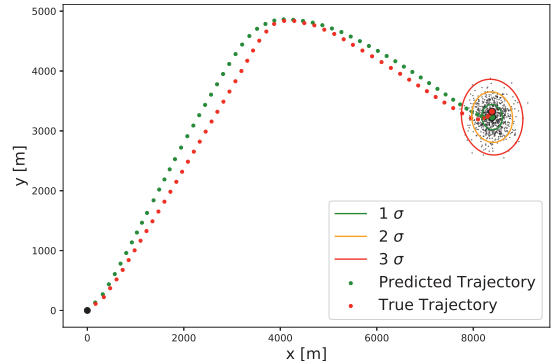


Fig. 6. Prediction of complex trajectory. Contours are illustrated for $T_p = 30$.

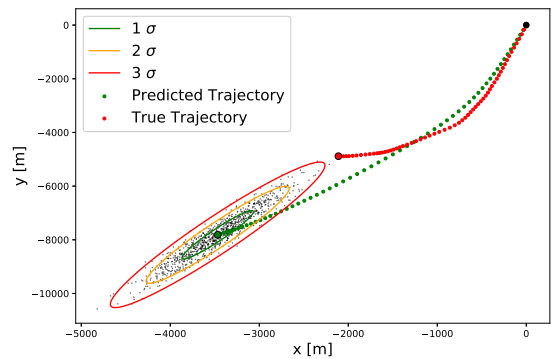


Fig. 7. Trajectory prediction with incorrect classification. Contours are illustrated for $T_p = 30$.

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, μ_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

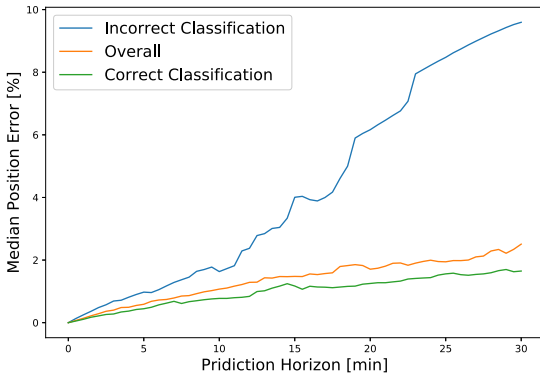


Fig. 8. Median position error of 100 randomly selected vessels as a function of the prediction horizon. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

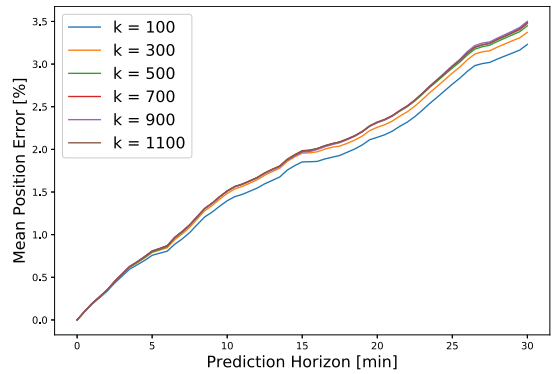


Fig. 10. Mean error for values of k used in interpolation.

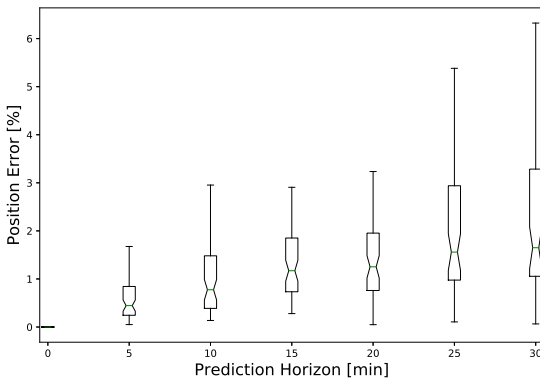


Fig. 9. Box plot of error for correctly classified vessels.

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 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 $\hat{h}_{f,s}$ and $\Sigma_{h,s}$.

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

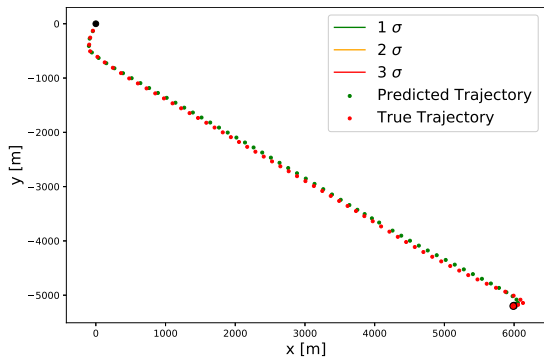


Fig. 11. Prediction of ferry trajectory. $T_p = 30$.

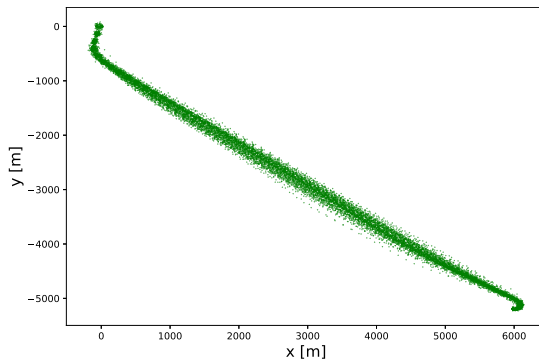


Fig. 13. Cluster used for ferry trajectory prediction.

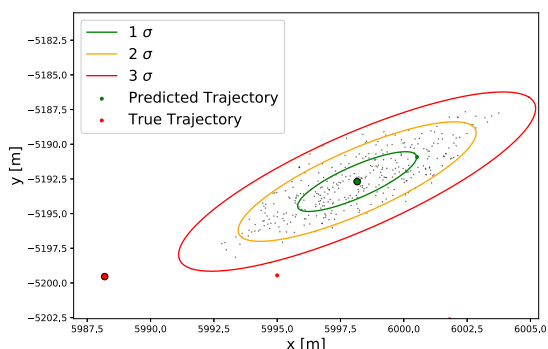


Fig. 12. Uncertainty contours for ferry trajectory prediction.

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

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%

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.

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 extracted 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

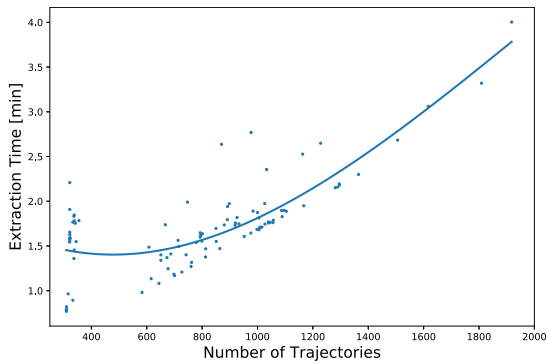


Fig. 14. Trajectory extraction running time.

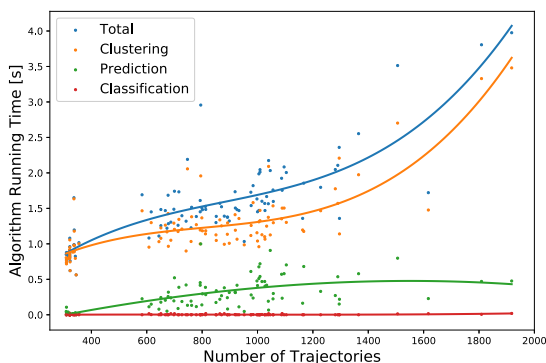


Fig. 15. Algorithm running times.

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 to 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, Beijing, 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>.

Paper III

Unsupervised Trajectory Anomaly Detection for Situation Awareness in Maritime Navigation

Brian Murray and Lokukaluge Prasad Perera (2020)

Published in *Proceedings of the 39th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2020)*. ASME.

<https://doi.org/10.1115/OMAE2020-18281>

OMAE2020-18281

UNSUPERVISED TRAJECTORY ANOMALY DETECTION FOR SITUATION AWARENESS IN MARITIME NAVIGATION

Brian Murray*

Department of Technology and Safety
UiT The Arctic University of Norway
9037 Tromsø, Norway
Email: brian.murray@uit.no

Lokukaluge P. Perera

Department of Technology and Safety
UiT The Arctic University of Norway
9037 Tromsø, Norway
Email: prasad.perera@uit.no

ABSTRACT

Situation awareness is essential in conducting effective collision avoidance in potential ship encounter situations. It has been shown that data driven trajectory prediction techniques, utilizing historical AIS data, have the potential to aid in providing such awareness. However, such data driven techniques will not perform well for unusual ship behavior, i.e. anomalous trajectories. Additionally, such anomalies in the dataset can corrupt the predictions. In this study, an unsupervised approach to anomaly detection is presented to aid such trajectory predictions. Gaussian Mixture Models are used to cluster trajectories, such that clusters of both normal and anomalous trajectories are discovered. Further, anomalies are discovered within clusters of normal behavior. Novel trajectories can then also be evaluated based on a parametric description of the historical ship traffic. The approach is shown to be effective in detecting anomalies relevant in such a trajectory prediction scheme.

NOMENCLATURE

a	Arbitrary AIS Parameter Vector
A	Set of AIS Data
b	Backward Trajectory Feature Vector
C	Data Cluster
d	Mahalanobis Distance
e	Eigenvector
E	Eigenvector Matrix

h	Latent Feature Vector
H	Distribution of Latent Features
j	Matrix Position
l	Number of Eigenvectors
L	Number of Data Points in Selected Trajectory
N	Number of Trajectories
M	Number of Models in GMM
s	Vessel State Vector
T	Time Horizon [min]
v	Speed over Ground [m/s]
x	UTM x-coordinate [m]
y	UTM y-coordinate [m]
z	Orthogonal Space
Λ	Eigenvalue Matrix
μ	Mean Vector
π	Prior Distribution
σ	Standard Deviation
Σ	Covariance Matrix
χ	Course over Ground [°]
<i>Subscripts</i>	
0	Initial State
b	Backward Trajectory
i	Sample Number
m	Model Number in GMM
δ	Maximum Offset
<i>Acronyms</i>	
AIS	Automatic Identification System
GMM	Gaussian Mixture Model
KL	Karhunen-Loève

*Address all correspondence to this author.

INTRODUCTION

The ubiquity of sensor data, in addition to enhanced computational power, has opened the door for machine learning techniques to be applied across a number of industries. Many argue that we are currently undergoing a fourth industrial revolution, often referred to as Industry 4.0 [1]. The automotive industry for instance has been active in adopting modern technologies, and integrating them into their designs. The shipping industry, however, has historically been a bit more conservative in adopting new technology. Nonetheless, the industrial revolution has reached the shipping industry as well, with some calling it Shipping 4.0 [2]. The purpose of incorporating such technologies is to enhance the efficiency and safety of ship operations. With respect to safety, effective collision avoidance is arguably among the most essential aspects of safe maritime operations, and depends on the level of situation awareness the navigator or system has. Situation awareness is often defined as having an awareness of what is going on around you, and the potential implications for your present and future situation [3].

In [4], historical AIS data are utilized to aid in providing situation awareness in future vessels. The goal is to predict the future trajectory of a selected vessel, such that collision avoidance maneuvers can be made far in advance of a potential encounter situation. Unusual ship behavior, i.e. anomalous trajectories, that do not correspond to regular behavior for the region, will be difficult to predict, and the algorithm will undoubtedly perform poorly. It is therefore of interest to identify anomalous ship behavior, such that the navigator can be made aware of the algorithm's inability to effectively predict the future trajectory of that particular vessel. Additionally, such predictions depend on the data that the algorithm is trained on. It is, therefore, of interest to identify anomalous historical AIS trajectories that can corrupt the predictions. Once identified, such anomalies can be removed from the dataset utilized in the trajectory prediction. The term anomaly is used in various settings in the maritime domain, where domain experts may disagree on the definition [5]. In this study, therefore, a trajectory anomaly is defined as behavior that is inconsistent with normal traffic to a degree that it may compromise a trajectory prediction.

Related Work

Anomaly detection is a widely researched field encompassing a variety of domains. [6] discusses various techniques used to facilitate anomaly detection that are generally applicable in any domain. Extensive work has also been done specifically on maritime anomaly detection. [7] provides an overview of state-of-the-art research into maritime anomaly detection, where the authors categorize the types of trajectory anomalies as positional, contextual, kinematic, complex or data-related. Generally, most methods try to extract the normal behavior of vessels, and then classify new behavior as anomalous or not based on the descrip-

tion of the normalcy. In this study, the normalcy with respect to both positional and kinematic data are of interest.

Most methods utilize parametric or non-parametric models to model the normalcy of ship traffic. [8] uses a parametric approach with a Gaussian Mixture Model (GMM) [9] by separating regions into cells. Local probability density functions in each cell evaluate the location, speed and course over ground data as anomalous or not. [8] further compares the GMM approach with a non-parametric Kernel Density Estimation approach also used in [10] to model normalcy. These methods, however, do not consider relationships between data points in a trajectory, and rather examine whether or not a specific part of the trajectory is anomalous to that of the behavior of that region. [11] presents the Traffic Route Extraction and Anomaly Detection (TREAD) method which clusters entry points, way-points and stationary points of trajectories within a defined region using Density-Based Clustering of Applications with Noise (DBSCAN). Once clusters are identified, Kernel Density Estimation is utilized to model the data, and a sliding window technique is used for anomaly detection. Identifying clusters that represent normal behavior for a given traffic route is a powerful tool for anomaly detection. In the TREAD approach, however, solely classifying entry, end and way-points can result in large differences in sub-trajectories that are classified together, degrading the effectiveness of the clustering technique.

Contribution

The objective of anomaly detection in this study is to enhance trajectory predictions. As such, it is designed with respect to the architecture outlined in [4]. This study uses a parametric approach to trajectory anomaly detection. Clustering is utilized to group historical AIS ship trajectories in a given region, where the discovered clusters either represent groups of normal or abnormal behavior. By discovering such anomalous trajectories, one can conduct an effective preprocessing of the data, such that a subsequent trajectory prediction technique can be run on an enhanced dataset. Additionally, the method can be utilized to classify an observed trajectory as anomalous based on the historical behavior of ships in the same region. If an observed trajectory is classified as anomalous, the prediction algorithm will not perform well, given that it is biased towards regular ship behavior. As such, the user can be made aware of the anomalous behavior, in addition to the likely limited predictive ability of the algorithm.

In this study, only relevant trajectories are extracted, that represent either where the region the vessel has been in the past, or may be in the future, relative to the observed state of the selected vessel. As such, the method is not general for any ship in the area, but selects relevant data for each individual case. Trajectories are evaluated in their entirety via a spectral anomaly detection approach [6]. In this manner, all parts of the trajectory

are represented in a low dimensional space, that more easily facilitates anomaly detection. Similarly to other approaches, the method utilizes clustering facilitated via a GMM to represent the normalcy of the trajectories. However, the clustering does not take place in latitude-longitude space, but in the lower dimensional representation of the trajectories. As such, entire trajectories are clustered together such that a high level of discrimination is achieved, also with respect to sub-trajectories. The study further presents a method to utilize the flexibility of the GMM to identify clusters of anomalies. Anomalies within clusters representing normal behavior are also discovered by investigating various low dimensional representations of the data. These are also compared to anomalies discovered by investigating the Mahalanobis distance [12].

METHODOLOGY

The overall architecture of the method is illustrated in Figure 1. The algorithm begins with a global anomaly detection step, followed by clustering and intermediate anomaly detection steps. Subsequently, the output of these steps is fed into a local anomaly detection step. The detected trajectories are then removed from the dataset, such that only normal trajectories remain. Given the enhanced dataset, the same techniques can be also utilized to evaluate if an observed trajectory is anomalous or not.

Trajectory Extraction

Historical AIS trajectories are extracted based on the methodology outlined in [4]. In order to predict the future trajectory of a selected vessel, the method extracts relevant historical AIS trajectories. The initial state of the selected vessel, \mathbf{s}_0 is defined in (1):

$$\mathbf{s}_0 \rightarrow [x_0, y_0, \chi_0, \nu_0] \quad (1)$$

Given the initial state of the selected vessel, the method identifies historical ships that were at a similar position with a similar speed and course over ground, and defines them as an initial cluster \mathbf{C}_0 in (2):

$$\mathbf{C}_0 = \{ \mathbf{a}_i \in \mathbf{A} : (|x_{z'i} - x_{z'0}| \leq \delta_W \wedge |y_{z'i} - y_{z'0}| \leq \delta_H) \wedge (|\chi_i - \chi_0| \leq \chi_\delta \wedge |\nu_i - \nu_0| \leq \nu_\delta) \} \quad (2)$$

Given \mathbf{C}_0 , unique trajectories are identified, with the point closest to \mathbf{s}_0 in $[x, y]$ -space defined as the initial state of each trajectory. \mathbf{C}_0 is then updated to solely contain the initial states of the discovered trajectories. Subsequently, the backward, i.e. past, trajectories are extracted relative to the the initial states

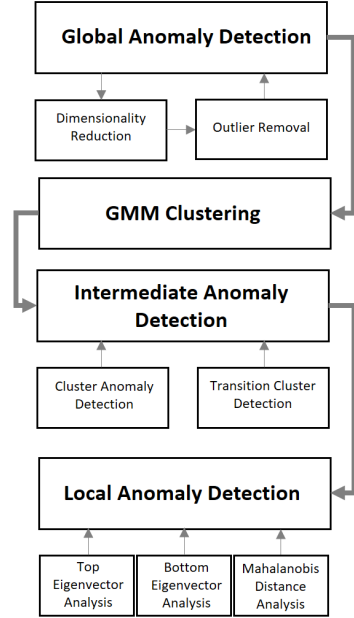


FIGURE 1. METHOD ARCHITECTURE.

of each trajectory. In this study, it is assumed that the past 10 minutes of trajectory data are available for the selected vessel, either via real-time AIS streams, or other on-board sensor systems. Therefore, the backward trajectories are extracted with a duration of $T_b = 10$. Each trajectory is then interpolated at 30 second intervals such that each trajectory has $L = 20$ entries. Each backward trajectory is then defined as a feature vector \mathbf{b} containing the flattened spatial data, $[x, y]$. By investigating these features, one can compare whether or not an observed trajectory is anomalous. Alternatively, the forward, i.e. future, trajectories can also be extracted to facilitate preprocessing for a subsequent prediction of the selected vessel's future trajectory. This prediction approach is outlined in [4]. In this study, however, solely the backward trajectories are investigated to evaluate the anomaly detection methodology.

Global Anomaly Detection

The first phase of the anomaly detection methodology is to identify outliers in the dataset. These are considered global anomalies, as they are anomalies with respect to the entire dataset, and not local anomalies with respect to the clusters of which it consists. These anomalous trajectories are identified and removed from the dataset, such that the subsequent clustering is more effective.

Dimensionality Reduction. To provide a basis for anomaly detection, a dimensionality reduction technique is applied to the trajectory feature vectors. This is often referred to as spectral anomaly detection [6]. Such methods assume that the lower dimensional space will yield a representation where normal and anomalous data can be separated. Dimensionality reduction can be facilitated via the Karhunen-Loève (KL) transform [13] in (3), where the dimensionality is reduced from $2L$ to l . \mathbf{E} is a matrix containing the top eigenvectors of the data's covariance matrix as its columns, in order of the magnitude of the corresponding eigenvalues.

$$\mathbf{h}_b = \mathbf{E}^T \mathbf{b} \quad (3)$$

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

$$\mathbf{\Sigma} = \mathbf{E} \mathbf{\Lambda} \mathbf{E}^T \quad (4)$$

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

Outlier Removal. In this study, it is suggested to reduce the dimensionality of the data such that $l = 2$. In this manner, the data will be projected onto a subspace spanned by the top two eigenvectors of the covariance matrix. In this subspace, it is assumed that the data can adequately be described by a Gaussian distribution, \mathbf{H}_b . A multivariate Gaussian distribution is therefore fit to the data by using the maximum likelihood estimates of its mean and covariance according to (6) and (7) respectively.

$$\mathbf{H}_b \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}) \quad (5)$$

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^N \mathbf{h}_{b,i}}{N} \quad (6)$$

$$\mathbf{\Sigma} = \frac{\sum_{i=1}^N (\mathbf{h}_{b,i} - \boldsymbol{\mu})(\mathbf{h}_{b,i} - \boldsymbol{\mu})^T}{N - 1} \quad (7)$$

Given that the data is described by a bivariate Gaussian distribution, the ellipse corresponding to the 3σ confidence interval can be determined. Anomaly detection can then be facilitated by thresholding points outside a value of 3σ . This technique is often used in the quality control domain [14]. All points that are outside the 3σ ellipse are, therefore, defined to be outliers, and labeled as anomalous trajectories. These trajectories are removed from the dataset.

Intermediate Anomaly Detection

In this section, unsupervised learning is utilized to cluster the extracted historical trajectories. Each resultant cluster represents a group of trajectories that have a high degree of similarity. The method is able to discover clusters that represent regular ship behavior, i.e. regular ship routes, in addition to clusters of anomalous trajectories. These clusters are defined to be either anomaly clusters, or transition clusters that represent trajectories between regular ship routes. These, however, are anomalies of the regular routes, and the discovered transition clusters are in fact clusters of anomalies as well.

Clustering is a technique that groups data together based on a distance measure. Data points that are closer to each other have a higher degree of similarity, and will as such be more likely to be part of the same group. However, in high-dimensional spaces, clustering algorithms may suffer due to the curse of dimensionality [15]. One issue relating to the curse of dimensionality is that as the number of dimensions increases, data points become increasingly sparse. As such, data points may have large distances between them with respect to certain dimensions. Clustering algorithms will, therefore, struggle to find groups of data in such high dimensional spaces. As a result, it is suggested to reduce the dimensionality of the dataset prior to clustering.

The suggested dimensionality reduction technique, prior to clustering, is the KL-transform in (3). This is conducted on the improved dataset, i.e. after outlier removal. In this section, however, l is not fixed, but adapts to the underlying data such as to preserve an adequate amount of data. l is, therefore, chosen such that the data compression preserves at least 99% of the data. This is facilitated by evaluating the sum of the eigenvalues for the top l eigenvectors as a percentage of the sum of all eigenvalues. By varying l one can determine the necessary number of eigenvectors to include in \mathbf{E} in (3).

Gaussian Mixture Model Clustering. Once the dimensionality reduction operation is conducted, clustering is facilitated via a GMM. A GMM assumes that the underlying data is comprised of M Gaussian distributions. Each Gaussian distribution has its own mean $\boldsymbol{\mu}_m$, covariance $\mathbf{\Sigma}_m$ and prior distribution $\boldsymbol{\pi}_m$. The algorithm begins by randomly assigning the mean values to be random data points in the distribution, and initializes the covariances as the identity matrix. The Expectation Maximization algorithm [9] is then utilized to update the model parameters such that the mixture model best fits the underlying data. The number of underlying distributions, M , i.e. the number of clusters, can also be unspecified, where the most likely number of clusters is determined by investigating the Bayesian Information Criterion. Trajectories are then classified to the Gaussian distribution of highest probability according to Bayes rule, resulting in M clusters of data. For more information on the GMM trajectory clustering methodology, see [4].

Cluster Anomalies. Given the flexibility of the GMM to adapt to the underlying data, it is able to identify clusters of both regular ship behavior, as well as clusters of anomalous behavior. The first type of such an anomaly grouping is defined as being a cluster anomaly, i.e. the entire cluster is classified as an anomaly. This is facilitated via a similar method as that outlined to detect global anomalies. By only considering the data along the top two eigenvectors, i.e. $l = 2$, a new Gaussian distribution is fit to the data in this two-dimensional subspace according to (5)-(7). A cluster is then considered anomalous if its mean, i.e. μ_m , lies outside the ellipse representing the 3σ confidence interval.

Transition Cluster Anomalies. The second type of anomaly grouping is referred to as a transition cluster in this study. These clusters can not be thresholded according to their means as they lie within the 3σ confidence interval of the dataset. Such clusters, however, lie between clusters of regular behavior, and in this view represent transitions between the various routes discovered. As such, these clusters contain outliers of the neighboring clusters of regular behavior. Determining if a cluster is a transition cluster is, however, more challenging, as one cannot threshold based on their location in the subspace. However, one can investigate the scatter of the clusters. A cluster that represents regular ship behavior will be more compact and have a higher density than that of a transition cluster. As such, clusters that have a high degree of variance are more likely to be a cluster of anomalous trajectories. This study therefore suggests to threshold clusters based on the trace of the covariance matrix of each cluster. The trace will give a measure of the variance of the cluster, and can be thresholded based on the standard deviation of the traces of all clusters in the GMM. Therefore, if the trace of a given cluster is outside 1σ from the mean, it is defined as a transition cluster containing anomalous trajectories.

Local Anomaly Detection

Once the anomaly and transition clusters are identified, only the clusters of regular, i.e. non-anomalous, ship trajectories remain. Each of these remaining clusters represent trajectories that spatially are quite similar. However, there may exist anomalous trajectories within these clusters as well. These are not captured by the previous techniques. Such trajectories will generally have the same pattern as that of the prevailing route, but contain inconsistencies that can be categorized as anomalous behavior. In this section, each regular cluster is investigated individually to identify anomalies of the cluster itself. If the tested trajectory is an outlier of all discovered clusters, it is classified as an anomaly. Additionally, all trajectories identified as belonging to an anomaly or transition cluster should be subjected to local anomaly detection, as they may in fact belong to a cluster of normal behavior, but were classified to an anomalous cluster along

the boundary between classes. These trajectories are referred to as local anomalies in this study.

Top Eigenvector Analysis. One technique to discover local anomalies it to utilize a method similar to that utilized for global anomaly detection. In this case, the KL-transform is first utilized to reduce the dimensionality of the trajectories. The largest degree of variation in the dataset will be captured by the projection of the data onto the top eigenvectors of the covariance matrix. Similarly to global anomaly detection, the cluster is assumed to be adequately described by a Gaussian distribution. This is arguably an adequate assumption as the cluster is created via a GMM. As for the global anomalies, the data is projected onto the top two eigenvectors of the covariance matrix. The mean, μ , and covariance, Σ , are then calculated in this subspace according to (6) and (7). In the same manner as for the global anomalies, the 3σ ellipse can then be evaluated. Any data points outside the ellipse are classified as anomalies. Additionally, the 2σ ellipse can be evaluated to increase the number of detected anomalies.

Bottom Eigenvector Analysis. Despite the top eigenvectors containing the most information in the cluster, they do not necessarily capture the necessary information to detect all anomalous trajectories. [16] found that also investigating the bottom eigenvectors, i.e. those with the lowest degree of variance, gave better results for anomaly detection. The same method as described for the top eigenvectors can then be utilized to threshold data outside the 2σ and 3σ ellipses.

The trajectory data will be quite compact when projecting it onto the bottom eigenvectors, and potential anomalies should be easily detected. These anomalies, however, will correspond to regions in the trajectories where there is generally a high degree of similarity. It is therefore likely that these anomalies represent anomalous sub-trajectories within the cluster.

Mahalanobis Distance Analysis. When investigating the top and bottom eigenvectors, one looks only in two directions at a time. As a result, anomalies that lie along the projection of the data onto the remainder of eigenvectors are not considered. The Mahalanobis distance gives a measure of the distance a data point is from the mean of the distribution. The metric projects the data onto the eigenvectors of the covariance matrix, but normalized such that they each have unit variance in the transformed space. One can in this manner consider the aforementioned 3σ ellipses as being described by a constant Mahalanobis distance if the data were solely two-dimensional. In this manner, the Mahalanobis distance takes into consideration all eigenvectors of the covariance matrix in a single distance metric.

In the case that the data is described by a multidimensional

Gaussian distribution, the probability density of an observation is related to the Mahalanobis distance. It can be shown that the squared Mahalanobis distance of the data, shown in (8), is χ^2 -distributed. The χ^2 -distribution, however, depends on the number of degrees of freedom of the data. In this case, the data will have $2L$ degrees of freedom, as no dimensionality reduction has taken place. The p-value of a given observation can be determined by using the value of χ^2 cumulative distribution function, with $2L$ degrees of freedom. If a trajectory has a p-value less than a given threshold, it is classified as an anomaly. In this study a p-value of 0.05 is investigated.

$$d^2 = (\mathbf{b} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{b} - \boldsymbol{\mu}) \quad (8)$$

RESULTS AND DISCUSSION

In this section, the outlined ship trajectory anomaly detection method is evaluated with respect to a case, in order to illustrate its performance. The investigated case corresponds to a region outside the city of Tromsø, Norway. A randomly selected data point from the historical AIS database corresponding to ship trajectories from January 1st 2017, to January 1st 2018 was utilized. The results illustrate how anomaly detection can be utilized to facilitate preprocessing for a subsequent trajectory prediction of the selected vessel. Similarly, the same techniques can be utilized to classify the behavior of the selected vessel as anomalous or normal. Global anomalies are first identified for the selected case, before potential transition and anomaly clusters are identified. Subsequently, one of the discovered clusters of normal behavior is investigated to illustrate local anomaly detection. The coordinate system of the figures in this section, with respect to $[x, y]$, is defined as having an origin at $[x_0, y_0]$ to better illustrate the scales involved.

Global Anomaly Detection Results

In this section, the results from the global anomaly detection are presented. The discovered anomalies correspond to the red trajectories illustrated in Figure 2. For this case, it appears that the global anomalies are comprised primarily of vessels that have a significant change in heading, i.e. those at the bottom of the figure, as well as vessels that have unusually high speeds, i.e. those in the upper left and lower right. By removing these trajectories from the dataset, a new covariance matrix can be calculated based on the remaining data, providing a much better basis for the subsequent cluster analysis.

Intermediate Anomaly Detection Results

In this section, the results from the intermediate anomaly detection are presented. The results of the clustering via a fitted

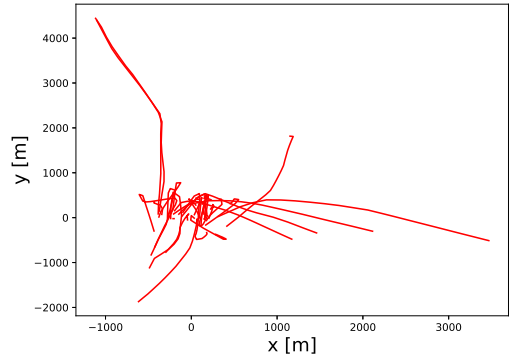


FIGURE 2. ANOMALOUS SHIP TRAJECTORIES IDENTIFIED VIA GLOBAL ANOMALY DETECTION.

GMM, are presented in Figure 3, where all extracted trajectories after removing the global anomalies are illustrated. Each color indicates a unique cluster. In this case, five clusters were discovered. The GMM was fit in the subspace comprised of the top four eigenvectors of the covariance matrix, as these accounted for more than 99% of the information in the data. The clustering is effective was this case, where unique ship routes were discovered.

The projection onto the top two eigenvectors is visualized in Figure 4. The red ellipse illustrates the 3σ confidence interval, and the colored dots correspond to each cluster. Each cluster is illustrated by a unique color. The large circles illustrate the mean of each Gaussian distribution comprising the GMM. By thresholding the means according to the 3σ ellipse, the light green cluster was identified as an anomaly cluster. The trajectories belonging to this cluster are illustrated in Figure 5. It is clear that the trajectories are erratic and lack any general pattern. Additionally, by evaluating the trace of the covariance matrix of each cluster, the purple cluster was classified as a transition cluster. This is evident from Figure 4, as the data is very sparse. The transition cluster is illustrated in 6. The purple cluster appears to contain outliers of all the others, where trajectories are in fact a transition between the other discovered clusters. In this manner, the GMM has been able to discover entire clusters of anomalies.

Figure 3 also contains the clusters of normal trajectory behavior. On the left of the figure, two clusters are visible, where the blue represents a main cluster of ship trajectories, with a dense yellow cluster within the same region. Both clusters appear to contain trajectories of normal behavior. On the right, the dark green cluster also appears to contain regular trajectories.

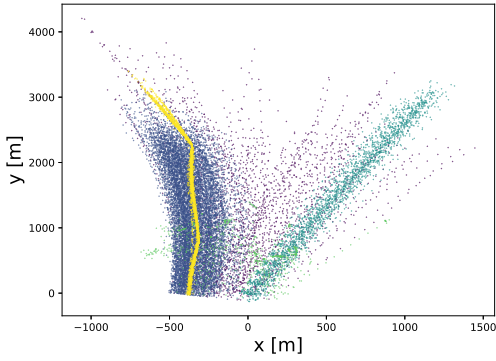


FIGURE 3. CLUSTERED SHIP TRAJECTORIES AFTER REMOVING GLOBAL ANOMALIES.

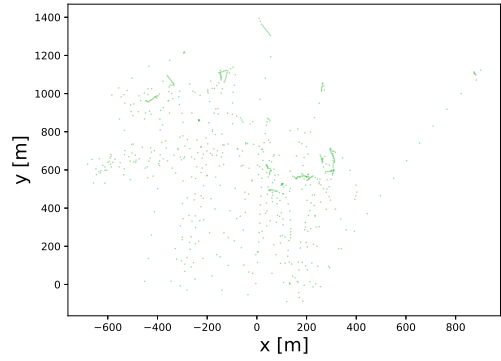


FIGURE 5. SHIP TRAJECTORIES CORRESPONDING TO ANOMALY CLUSTER.

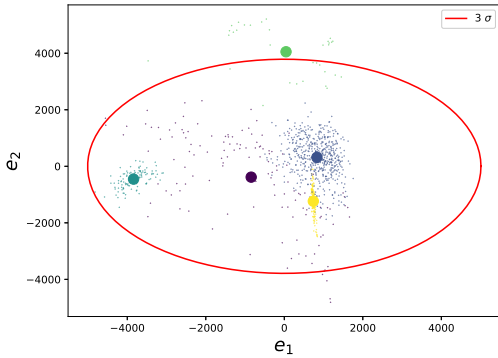


FIGURE 4. TOP EIGENVECTOR REPRESENTATION OF TRAJECTORY CLUSTERS. TRANSITION AND ANOMALY CLUSTERS ARE IDENTIFIED IN THIS SUBSPACE.

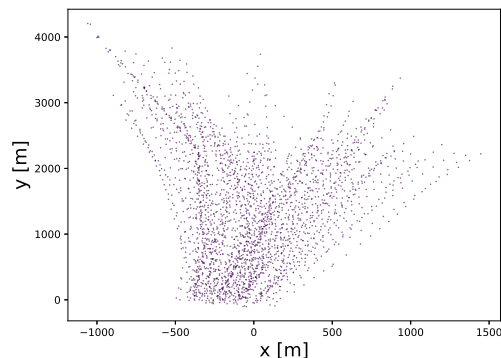


FIGURE 6. SHIP TRAJECTORIES CORRESPONDING TO TRANSITION CLUSTER.

Local Anomaly Detection Results

In this section, the results from the local anomaly detection are presented. To evaluate the performance of the technique, one cluster was investigated. The case presented in this section is that of the blue cluster discovered by the GMM. Anomalies were discovered by analyzing the top and bottom eigenvectors in addition to the Mahalanabis distance.

Top Eigenvector Analysis Results. Figure 7 illustrates the results from the top eigenvector analysis. The orange and red trajectories are those classified as anomalies, i.e. they

fell outside the 2σ and 3σ thresholds, respectively. It is clear that reducing the threshold increases the number of discovered anomalies significantly. 7 anomalies were detected out of 536 trajectories for threshold of 3σ , and 66 for 2σ . As such, many anomalous trajectories are not discovered if one thresholds based on the 3σ confidence interval. Given that the anomaly detection is meant to facilitate effective trajectory prediction, it may be of interest to reduce the threshold and filter out more unusual behavior. However, reducing the threshold may result in the classification of normal behavior as anomalous, falsely warning that the algorithm will not produce an accurate prediction.

An investigation was also conducted to interpret why these

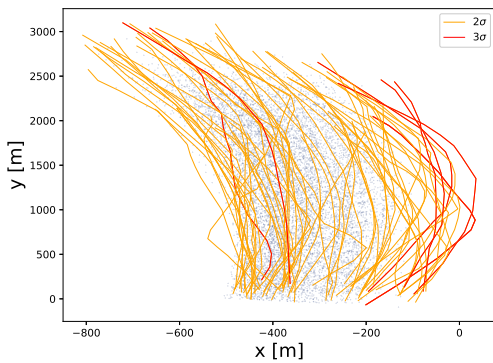


FIGURE 7. LOCAL ANOMALY DETECTION FROM TOP EIGENVECTOR ANALYSIS. THE ORANGE AND RED TRAJECTORIES CORRESPOND THE DETECTED ANOMALIES.

trajectories were classified as anomalies. This was facilitated by looking at the output of an element-wise multiplication of the top eigenvectors with the trajectory feature vectors. The relative importance of each position along the trajectory could then be evaluated with respect to its contribution in being classified as an anomaly or not. Figure 8 illustrates the relative weight contributing to anomaly classification, where the values are normalized by the maximum value of the contributions. The higher the value, the more the position contributes to anomaly classification. In this case, it appears that the top eigenvectors have focused on areas where ships have sailed outside the densest regions. It appears that the highest contributions come from the top left of the figure, i.e. ships that have sailed more quickly than normal, and center-right in the figure, i.e. ships that have sailed outside the normal region. All discovered trajectories, are, however, generally smooth.

Bottom Eigenvector Analysis Results. Figure 9 illustrates the detected anomalies from the bottom eigenvector analysis. This is facilitated in the same way as for the top eigenvectors. 14 anomalies were detected out of 536 trajectories for a threshold of 3σ , and 74 for 2σ . It is evident from the figure that new anomalies are detected compared to those for the top eigenvectors. The detected anomalies are not as smooth as those for the top eigenvectors, and have more irregular behavior within the trajectories themselves, i.e. sub-trajectories are anomalous. The overall shapes of the trajectories do not, however, appear anomalous. The top eigenvectors focused on the largest differences in the dataset. As a result, it is seen from Figure 7 that trajectories that fall outside the densest regions are detected. For the bottom

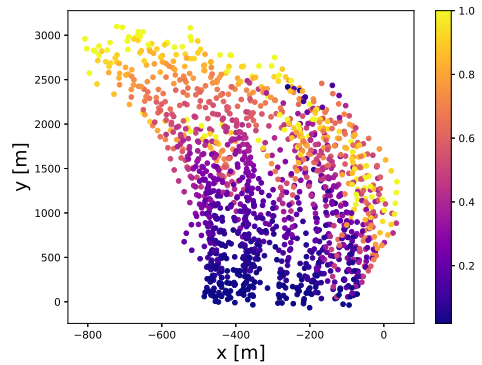


FIGURE 8. LOCAL ANOMALY DETECTION FROM TOP EIGENVECTOR ANALYSIS. THE COLOR CORRESPONDS TO RELATIVE WEIGHT CONTRIBUTING TO ANOMALY CLASSIFICATION.

eigenvectors, however, the subspace describes the least variation in the data, i.e. the distribution will be dense. As such, most trajectories should be similar in this subspace.

Figure 10 illustrates the relative weight contributing to anomaly classification for the bottom eigenvector analysis in the same manner as for the top eigenvectors. It is clear here that the method focuses on different regions of the trajectories as compared to the top eigenvectors. It appears that irregular sub-trajectories contribute more in this case. As such, it appears that the top and bottom eigenvectors discover different types of anomalies, both of which are relevant for detection.

Mahalanobis Distance Analysis Results. The Mahalanobis distance metric differs from the top and bottom eigenvector analyses as it takes into consideration the normalized projection onto all eigenvectors of the data. Figure 11 illustrates the discovered anomalous trajectories. It is clear from the figure that far more anomalies were discovered compared to the aforementioned methods. 84 out of the 536 trajectories in the cluster were identified as anomalies in this case. When investigating the data, it appears that the method discovers a combination of anomalies detected by both the top and bottom eigenvectors, in addition to new anomalies not detected by either subset. Furthermore, certain trajectories that were classified as anomalies in the previous techniques, are not detected via the Mahalanobis method.

The trajectories that are discovered via this technique appear to be irregular, where trajectories crossing across the main direction of the route, as well as those with irregular sub-trajectories, are detected. The Mahalanobis technique appears, therefore, to

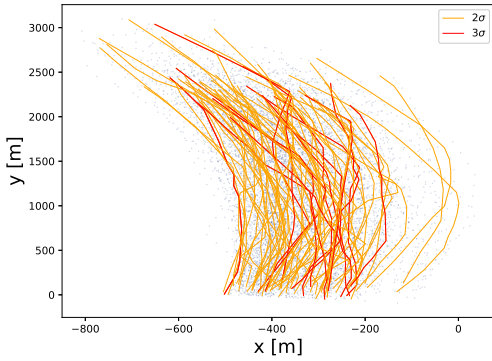


FIGURE 9. LOCAL ANOMALY DETECTION FROM BOTTOM EIGENVECTOR ANALYSIS. THE ORANGE AND RED TRAJECTORIES CORRESPOND TO THE DETECTED ANOMALIES.

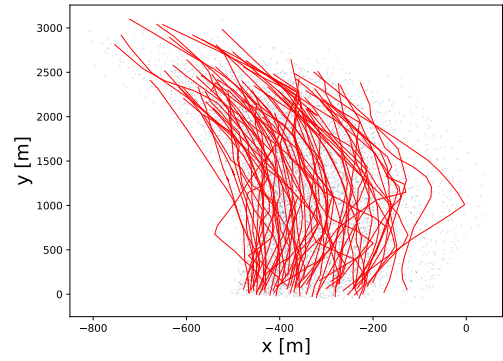


FIGURE 11. LOCAL ANOMALY DETECTION VIA MAHALANOBIS DISTANCE METRIC. THE RED TRAJECTORIES CORRESPOND TO THE DETECTED ANOMALIES.

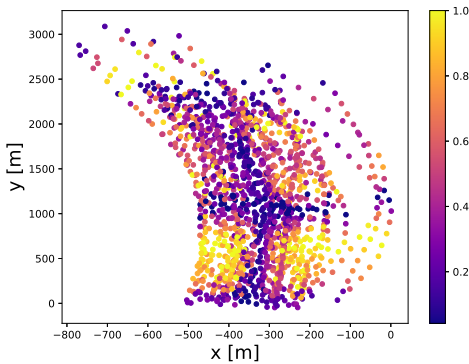


FIGURE 10. LOCAL ANOMALY DETECTION FROM BOTTOM EIGENVECTOR ANALYSIS. THE COLOR CORRESPONDS TO RELATIVE WEIGHT CONTRIBUTING TO ANOMALY CLASSIFICATION.

be quite powerful in discovering such anomalies within the cluster. This is likely due to the Mahalanobis distance taking into account all information in the trajectory, and not thresholding after dimensionality reduction.

As the metric normalizes the projections onto the eigenvectors, there will be no bias towards the projection onto specific eigenvectors, and anomalies along any eigenvector will have an effect on the distance metric. However, trajectories that generally appear normal, but are somewhat anomalous with respect to

many eigenvectors, may be misclassified as anomalies. This is due to the Mahalanobis distance taking contributions from all directions. As a result, normal behavior is more likely to be classified as anomalous with this approach than in the aforementioned eigenvector approaches. This appears to be supported by Figure 11, where some of the trajectories appear to be normal, likely contributing to the high number of detections. Nonetheless, the method is effective in detecting new anomalies not detected by the eigenvector approaches.

CONCLUSION AND FURTHER WORK

In this study, trajectory anomaly detection is facilitated with respect to aiding ship trajectory prediction. Based on a trajectory extraction scheme, relevant ship trajectories are represented by lower dimensional vectors. These trajectory representations are effective in discriminating between various groups of trajectories during clustering. By extracting trajectories based on a fixed length of time, speed differences are inherently preserved in the positional data. The GMM clustering, therefore, is effective in detecting normal ship routes with a high degree of fidelity. In addition, the flexibility of the technique allows it to discover clusters of anomalous data, which can be identified as either anomaly or transition clusters.

Local anomaly detection, within clusters of normal behavior, is shown to be achieved by investigating the top and bottom eigenvectors of the covariance matrix of the data. The projection onto these eigenvectors allows for ship trajectories with anomalous speeds, as well as unusual sub-trajectories, to be discovered. However, it is clear that different anomaly modes are discovered by investigating the various eigenvector projections, both of im-

portance for detection. Alternatively, the Mahalanobis distance allows for the discovery of entirely new anomalies. The method, however, also likely classifies some normal behavior as anomalous, and may therefore pose a challenge when using it to evaluate new observed trajectories. The outlined method, however, appears to facilitate effective preprocessing for trajectory prediction, as well as provide a method evaluate new ship behavior. Further work will investigate the sensitivity of thresholding in a two-dimensional subspace against utilizing the Mahalanobis distance on a subset of eigenvectors. Additionally, various thresholds will be investigated to evaluate the types of anomalies not detected by the current architecture.

ACKNOWLEDGMENT

This work was supported by the Norwegian Ministry of Education and Research and the MARKOM2020 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

- [1] Hermann, M., Pentek, T., and Otto, B., 2016. "Design Principles for Industrie 4.0 Scenarios". In 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, pp. 3928–3937.
- [2] Rødseth, Ø. J., Perera, L. P., and 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.
- [3] Endsley, M. R., Bolté, B., and Jones, D. G., 2003. *Designing for situation awareness : an approach to user-centered design*. Taylor & Francis.
- [4] Murray, B., and 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 (OMAE 2019).
- [5] Roy, J., 2008. "Anomaly detection in the maritime domain". In Optics and Photonics in Global Homeland Security IV, C. S. Halvorson, D. Lehrfeld, and T. T. Saito, eds., Vol. 6945, International Society for Optics and Photonics, SPIE, pp. 180 – 193.
- [6] Chandola, V., Banerjee, A., and Kumar, V., 2009. "Anomaly detection: A survey". *ACM computing surveys (CSUR)*, **41.3**(15).
- [7] Riveiro, M., Pallotta, G., and Vespe, M., 2018. "Maritime anomaly detection: A review". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **8**(5).
- [8] Laxhammar, R., Falkman, G., and Sviestins, E., 2009. "Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator". *12th International Conference on Information Fusion*(July), pp. 756–763.
- [9] Reynolds, D., 2015. *Gaussian Mixture Models*. Springer US, Boston, MA, pp. 827–832.
- [10] Ristic, B., Scala, B. L., Morelande, M., and Gordon, N., 2008. "Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction". *2008 11th International Conference on Information Fusion*, pp. 40–46.
- [11] Pallotta, G., Vespe, M., and Bryan, K., 2013. "Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction". *Entropy*, **15**(12), jun, pp. 2218–2245.
- [12] Mahalanobis, P. C., 1936. "On the generalized distance in statistics". National Institute of Science of India.
- [13] Karhunen, K., 1946. "Zur spektraltheorie stochastischer prozesse". *Annales Academiae Scientiarum Fennicae*, **37**.
- [14] Shewhart, W. A., 1931. *Economic control of quality of manufactured product*. Macmillan And Co Ltd, London.
- [15] Steinbach, M., Ertöz, L., and Kumar, V., 2004. "The Challenges of Clustering High Dimensional Data". In *New Directions in Statistical Physics*. Springer, Berlin, Heidelberg, pp. 273–309.
- [16] Parra, L., Deco, G., and Miesbach, S., 1996. "Statistical Independence and Novelty Detection with Information Preserving Nonlinear Maps". *Neural Computation*, **8**(2), pp. 260–269.

Paper IV

Deep Representation Learning-Based Vessel Trajectory Clustering for Situation Awareness in Ship Navigation

Brian Murray and Lokukaluge Prasad Perera (2021)

Accepted for Publication in *Developments in Maritime Technology and Engineering. Proceedings of the 5th International Conference on Maritime Technology and Engineering (MARTECH 2020)*. Taylor and Francis.

Deep Representation Learning-Based Vessel Trajectory Clustering for Situation Awareness in Ship Navigation

Brian Murray & Lokukaluge Prasad Perera

Department of Technology and Safety

UiT The Arctic University of Norway, Tromsø, Norway

ABSTRACT: Vessel trajectory clustering using historical AIS data has been a popular research topic in recent years. However, few studies have investigated applying deep learning techniques. In this study, deep representation learning is investigated for use in clustering historical AIS trajectories to provide insight into navigation patterns to support maritime situation awareness. A recurrent autoencoder and β -variational recurrent autoencoder are investigated to generate fixed size vector representations of the AIS trajectories. Subsequently, clustering is facilitated by applying the Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm to the representations. The method was tested on historical AIS data for a region surrounding Tromsø, Norway, with successful results. The results also indicate that the β -variational recurrent autoencoder was able to generate representations of the AIS trajectories that resulted in more compact clusters.

1 INTRODUCTION

Maritime situation awareness is an essential aspect of safe maritime operations. Recently, historical Automatic Identification System (AIS) data have been the subject of significant research to aid in intelligent navigation systems (Tu et al. 2017) that can support in providing maritime situation awareness. One area of interest has been identifying navigational patterns for various geographical regions based on historical ship behavior. Knowledge of such patterns can be useful for multiple purposes including anomaly detection, route prediction, path planning, collision avoidance and general maritime situation awareness for ship operators. Multiple studies have addressed identifying such patterns, where trajectory clustering is a central element. Clustering is a field of machine learning concerned with identifying groupings within a dataset. Aarsæther and Moan (2009), for instance, applied computer vision techniques to find groupings of trajectories, and Pallotta et al. (2013) introduced the Traffic Route Extraction and Anomaly Detection (TREAD) methodology. Murray and Perera (2020) also clustered trajectories to predict the future position of a vessel for collision avoidance purposes. Other relevant studies include Zhang et al. (2018) and Zhou et al. (2019).

Many of the studies in the literature apply machine learning to historical AIS data to achieve the desired effect. To effectively conduct clustering, a similarity metric must be calculated to compare the trajectories.

This can be challenging as the trajectories may be of variable length. This is not conducive with standard clustering techniques that require data points to be vectors of equal length. Various techniques and metrics have been applied to overcome this challenge, e.g. Zhou et al. (2019) which applied Dynamic Time Warping (DTW). These techniques attempt to generate a representation of the trajectories such that they can be compared against each other. Few studies, however, have investigated utilizing deep learning techniques in AIS trajectory clustering. Nguyen et al. (2018) introduced a multi-task deep learning framework based on a variational recurrent neural network trained on historical AIS data for a region. This framework can be used for multiple tasks including trajectory reconstruction and anomaly detection. However, the framework employs a 4-hot encoding of the data that reduces the resolution. Additionally, the architecture does not provide suitable trajectory representations for clustering. Yao et al. (2017), however, investigated AIS trajectory clustering based on deep representation learning. The outlined method gave good results in providing representations of the trajectories, with the clustering based on the deep representations outperforming other non-deep learning based metrics. The method, however, does not investigate more advanced deep learning architectures, and can, therefore, be further improved. Additionally, the study employs the k-means clustering algorithm, which likely reduces the clustering performance compared to non-parametric, density-based approaches that better han-

dle clusters of varying shape and size.

In this study, deep representation learning is investigated to facilitate historical AIS trajectory clustering. A method is presented where deep representations of trajectories are generated for a given geographical region. A recurrent autoencoder is compared with a more advanced architecture, the β -variational recurrent autoencoder. Using the representations from these architectures, an approach to cluster the trajectories is introduced using a density-based clustering approach that can adapt to a high number of clusters of variable density and shape. The method is applied to a test case, and the results indicate that deep learning can generate powerful representations that facilitate effective clustering. The focus of the study is on the ability of deep learning to generate meaningful representations, and as such aims to determine if such representations are appropriate for use in trajectory clustering.

1.1 Deep representation learning

One of the most powerful aspects of deep learning is its ability to learn meaningful features. Features in this case are a representation of the data, that in the case of deep learning, are learned via the training of the network. Most applications within deep learning deal with classification tasks, and in such cases these features are optimized such that they can be used to discriminate between classes. Deep learning can be split into two main groups, supervised and unsupervised learning. In supervised learning, class labels for each data point are available. In a classification setting, the accuracy of the network in correctly classifying the data is utilized to optimize the parameters in the network. Unsupervised learning on the other hand, deals with cases in which labels for the data are unavailable. In such cases, it is desirable to discover the underlying structure in the data. In machine learning, the task of finding underlying groupings in the data is known as clustering. Deep learning in and of itself does not present a method to cluster the data, but can be utilized in an unsupervised form to generate more meaningful representations, i.e. features, through which the structure of the data is more apparent. Using such new representations, conventional clustering techniques can be applied to the data.

A meaningful representation of the data should be one that preserves information. One architecture that can generate such representations is the autoencoder. The simplest form of an autoencoder is a multi-layer perceptron (Bourlard and Kamp 1988). However, alternative frameworks also exist that utilize either convolutional or recurrent layers. An autoencoder has as its objective to reconstruct the data input to the network. Autoencoders can be separated into two parts, an encoder and a decoder. The encoder produces a latent representation of the input, and the decoder then reconstructs the data from this latent representation.

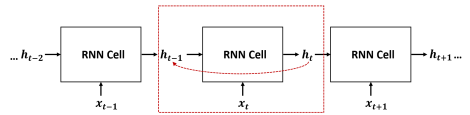


Figure 1: Unfolded RNN structure. The recurrent operation is illustrated within the red box.

It is desirable to learn meaningful representations in the latent space of the data. One approach to achieve this is to utilize undercomplete autoencoders (Goodfellow et al. 2016), i.e. where the latent space has a lower dimensionality than the input. Such architectures create a bottleneck in the latent space. In this manner, information must be compressed. By optimizing the network via the reconstruction loss, the network is forced to learn a meaningful representation of the data that preserves the mutual information between the input data and their latent representations. This data compression can also be viewed as a form of dimensionality reduction. It has been shown that deep autoencoders have the ability to produce much better representations of the data than other dimensionality reduction techniques such as Principle Component Analysis (PCA) (Hinton and Salakhutdinov 2006).

1.2 Recurrent neural networks

Historical AIS trajectories are datasets of multivariate time series. Traditional autoencoders utilize architectures that require a fixed size input for each data point. In the case of trajectory data, each data point is a time series of variable length. As such, in this study it is suggested to utilize the power of recurrent neural networks (RNNs) (Rumelhart et al. 1986) to facilitate effective trajectory representations.

RNNs are designed to process sequences, and are additionally capable of handling sequences of variable length. The networks are ideal for time series as they are able to incorporate temporal information and dependencies. In this sense, they have a form of memory. An RNN can be thought of as an unfolded computational graph, where the recurrent operation is unfolded. This is visualized in Fig. 1. Consider the sequence of length L : $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_L\}$. An RNN cell takes the current state of the sequence, \mathbf{x}_t as well as the previous hidden state \mathbf{h}_{t-1} as input. The cell then outputs the current hidden state, \mathbf{h}_t . This process then repeats for all states in the sequence. The output of the cell, i.e. the current hidden state, is fed back into the same cell along with the next state in the sequence. In this manner, the operation is recurrent, as visualized by the red box in Fig. 1. In this sense the same cell and parameters are shared between all operations.

The standard RNN, also known as the vanilla RNN, however, encounters challenges during training due to vanishing gradients during backpropagation (Bengio et al. 1994). This prevents the network from learning long-term dependencies. More advanced archi-

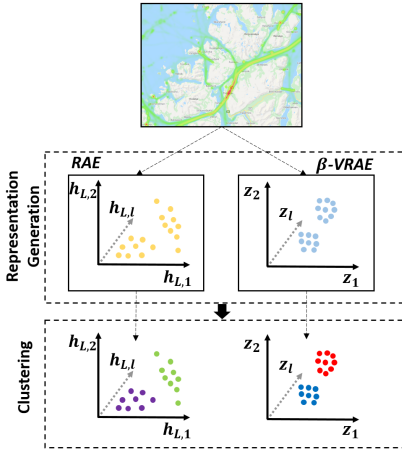


Figure 2: Overview of methodology. Map courtesy of Google Maps (2020).

tures have, therefore, been introduced to ameliorate the challenge of vanishing gradients. Gated recurrent architectures including Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (GRU) (Cho et al. 2014, Chung et al. 2014) address the issue of vanishing gradients through the introduction of gates. In this manner, the networks are able to learn long-term dependencies in the data.

2 METHODOLOGY

In this section, the methodology utilized to facilitate clustering of historical AIS trajectories is presented. The overall architecture is two-fold, and is illustrated in Fig. 2. In the first step, deep learning is utilized to facilitate representation generation. The architecture takes as input historical AIS trajectories for a given geographical region, and outputs a latent representation that can be further processed. In the second step, clustering is conducted on the latent trajectory representations to evaluate the ship traffic in the region. Two deep learning-based approaches are investigated in this study. The first is a recurrent autoencoder (RAE), and the second a β -variational recurrent autoencoder (β -VRAE). These approaches are further described in this section. Using the latent trajectory representations, the Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm is applied to cluster the trajectories. This study aims to investigate the ability of deep representation learning to facilitate effective clustering of vessel trajectories. As such, the study is limited to the methods described in this section.

2.1 Preprocessing

Prior to applying the representation generation step, a preprocessing of the the historical AIS data is conducted. Each unique vessel trajectory for the given re-

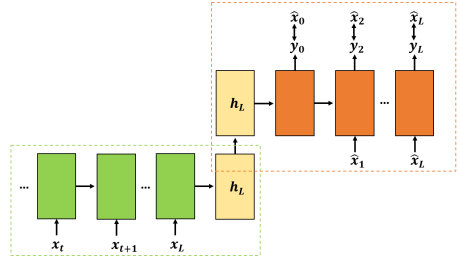


Figure 3: RAE architecture. The encoder is visualized in green, the final hidden state, i.e. latent representation, in yellow, and the decoder in orange.

gion is extracted and stored. Each trajectory is a sequence $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_L\}$ that has been interpolated at one minute intervals. Each state \mathbf{x}_t is a vector of spatio-temporal data, that includes the positional data x, y in UTM-coordinates, the speed over ground v , and the x - and y - components of the course over ground χ_x and χ_y in (1). Each of these values are normalized across the dataset to have values between -1 and 1 .

$$\mathbf{x}_t = [x, y, v, \chi_x, \chi_y] \quad (1)$$

2.2 Recurrent autoencoder

In order to learn good representations of the trajectories, an undercomplete autoencoder structure is investigated. This introduces a bottleneck through which the network must learn the best representation to reconstruct the data. Given that AIS trajectories are sequences of data, recurrent neural networks provide the core architecture applicable to generate meaningful representations. Some of the most popular forms of recurrent neural networks are sequence to sequence models (Sutskever et al. 2014). These provide the basis for many natural language processing tasks such as translation (Cho et al. 2014). The basis of these models is to train an encoder-decoder model in a similar manner to an autoencoder. The encoder takes an input sequence, for instance a sentence in English, and encodes it to a fixed size vector. The decoder then takes the hidden representation of the sentence and generates a target sentence in another language, for instance Spanish. The input and target sequences can be of variable length as well.

If one, however, utilizes a sequence to sequence model to reconstruct its input, i.e. with the target sequence equal to the input sequence, the architecture functions as a recurrent autoencoder (RAE) (Srivastava et al. 2015). The structure of an RAE is visualized in Fig. 3. In this case, the input sequence is run through an encoder recurrent neural network. The output of the final cell will have compressed the information in the sequence into a fixed size vector, i.e. the final hidden representation, \mathbf{h}_L . \mathbf{h}_L is then fed into a decoder recurrent neural network that predicts one time step at a time. First, the initial state \mathbf{x}_0 is predicted using \mathbf{h}_L as input. The following hidden state

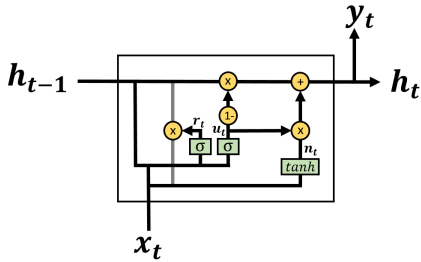


Figure 4: GRU cell. Illustration adapted from colah.github.io.

is then fed into the next cell along with the predicted initial state as input. Each predicted value, i.e. y_t , is estimated using a fully connected layer that takes the current hidden state, \mathbf{h}_t as input. This process then repeats for the remainder of the sequence. The network can be optimized by using the mean squared error as a loss function, where the error between the predicted and true states for each time step are calculated.

By training such an architecture, variable length AIS trajectories can be represented by a fixed size vector via their hidden representations, \mathbf{h}_L . This space is equivalent to the latent space for this architecture. When training the recurrent autoencoder, trajectories that have a higher degree of similarity will be in closer proximity to each other than dissimilar sequences in the latent space. Similarity measures can easily be evaluated based on distances in this space, and classical clustering techniques can be applied.

2.2.1 Gated recurrent unit

The more recent Gated Recurrent Unit (GRU) cell is a variant of the LSTM that reduces the number of parameters necessary to learn, and is, therefore, investigated for use in this study as the core architecture in the recurrent autoencoder. Each cell in Fig. 3 can, therefore, be thought of as a GRU cell. The architecture of the GRU cell is illustrated in Figure 4. The cell takes in the previous hidden state, \mathbf{h}_{t-1} , and the current input state, \mathbf{x}_t . The reset gate (2) and the update gate (3) regulate what information should be retained or forgotten in the network. Each gate comprises a weight matrix, \mathbf{W} , and bias term, \mathbf{b} , that consist of parameters that are updated during the training of the network. The output of the operations are fed into sigmoid activation functions. These force the values between 0 and 1, and are multiplied with the various inputs using the Hadamard product, thereby regulating the amount of information that should be passed on. In this manner, they function as gates, either allowing or preventing information from flowing. (4) calculates a new candidate vector for the hidden state via a hyperbolic tangent activation function. In (5), the hidden state is calculated and passed on to the next cell in the network. The hidden state, \mathbf{h}_t , can also be used for a prediction, \mathbf{y}_t , via (6).

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{b}_{xr} + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r) \quad (2)$$

$$\mathbf{u}_t = \sigma(\mathbf{W}_{xu}\mathbf{x}_t + \mathbf{b}_{xu} + \mathbf{W}_{hu}\mathbf{h}_{t-1} + \mathbf{b}_{hu}) \quad (3)$$

$$\mathbf{n}_t = \tanh(\mathbf{W}_{xn}\mathbf{x}_t + \mathbf{b}_{xn} + \mathbf{r}_t \odot (\mathbf{W}_{hn}\mathbf{h}_{t-1} + \mathbf{b}_{hn})) \quad (4)$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \mathbf{n}_t \quad (5)$$

$$\mathbf{y}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_{hy} \quad (6)$$

2.3 β -variational recurrent autoencoder

As previously mentioned, autoencoders can be powerful in generating meaningful representations of the data they are trained on. Typically, however, the latent space is sparsely populated. This is not an issue for tasks where compression is the goal of the autoencoder, as the scattered data indicate an ideal utilization of the latent space and often lead to better reconstruction results (Spinner et al. 2018). This, however, may be challenging for a clustering algorithm, as data points belonging to the same class may be scattered over a large region in the latent space. The variational autoencoder (Kingma and Welling 2014, Rezende et al. 2014) attempts to limit the chaos in the latent space. This is achieved by forcing latent variables, denoted \mathbf{z} , to become normally distributed. The main goal of the variational autoencoder is data generation, as they attempt to learn the underlying distribution of the data, $p(\mathbf{x})$, such that new data points can be generated by sampling from the distribution. However, the resulting latent representations of the data end up being more compact than for standard autoencoders, where similar data are more closely grouped. This may provide a better basis for a clustering algorithm.

The variational autoencoder is a probabilistic version of a traditional autoencoder. It is assumed that the data are generated by a random process utilizing a continuous random variable, \mathbf{z} . The general idea is that a value \mathbf{z}^i is generated from a prior distribution $p_\theta(\mathbf{z})$, and a data point \mathbf{x}^i is generated via some conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$.

The marginal likelihood $p_\theta(\mathbf{x})$ in (7) and posterior density $p_\theta(\mathbf{z}|\mathbf{x})$ in (8) are, however, intractable. As a result, the variational aspect of the autoencoder is introduced in that $p_\theta(\mathbf{z}|\mathbf{x})$ is replaced with an approximation, $q_\phi(\mathbf{z}|\mathbf{x})$. In the context of an autoencoder, the function $q_\phi(\mathbf{z}|\mathbf{x})$ can be thought of as a probabilistic encoder that produces a distribution over the latent variable, \mathbf{z} . $p_\theta(\mathbf{x}|\mathbf{z})$ can, therefore, be thought of as the decoder, taking \mathbf{z} , and reconstructing the input, \mathbf{x} .

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})d\mathbf{z} \quad (7)$$

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})} \quad (8)$$

It is assumed that the approximate posterior is a multivariate Gaussian with a diagonal covariance structure according to (11). A neural network is utilized to estimate the parameters of the distribution, i.e. the mean μ_z and standard deviation σ_z . The latent variable, \mathbf{z} , is then sampled from this distribution and decoded by $p_\theta(\mathbf{x}|\mathbf{z})$ to generate the reconstructed data, \mathbf{x} . A neural network is also used to model $p_\theta(\mathbf{x}|\mathbf{z})$. In Kingma and Welling (2014), a multi-layer perceptron network was suggested, but alternative architectures have also been proposed.

Fabius and van Amersfoort (2015) introduced the variational recurrent autoencoder (VRAE) by integrating an RNN architecture. As such, the encoder and decoder functions are assumed to be described by RNNs. In this study, a GRU architecture is utilized as the RNN cell. The encoder RNN produces the final hidden state \mathbf{h}_L , which compresses the information in the given AIS trajectory. The parameters of the normal distribution in (11) are subsequently calculated via fully connected layers in (9) and (10).

$$\mu_z = \mathbf{W}_\mu \mathbf{h}_L + \mathbf{b}_\mu \quad (9)$$

$$\sigma_z = \mathbf{W}_\sigma \mathbf{h}_L + \mathbf{b}_\sigma \quad (10)$$

$$q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mu_z, \sigma_z^2 \mathbf{I}) \quad (11)$$

However, training such an architecture using back-propagation is not possible, as gradients are unable to flow through the sampling operation. The reparametrization trick is, therefore, utilized. Instead of sampling from (11), the latent vector \mathbf{z} is calculated in (13). The sampling effect is achieved by sampling from a noise vector, ϵ , distributed according to (12). This allows for gradients to flow freely. The initial hidden state that is input to the decoder RNN is then calculated according to (14). Subsequently, the decoder RNN reconstructs the data in the same manner as for a conventional recurrent autoencoder. The overall architecture of the variational recurrent autoencoder is visualized in Figure 5.

$$\epsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (12)$$

$$\mathbf{z} = \mu_z + \sigma_z \odot \epsilon \quad (13)$$

$$\mathbf{h}_0 = \tanh(\mathbf{W}_{zh} \mathbf{z} + \mathbf{b}_{zh}) \quad (14)$$

The approximation of the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ by $q_\phi(\mathbf{z}|\mathbf{x})$ is optimized by maximizing a lower bound on the log-likelihood, i.e. $\log(p_\theta(\mathbf{x})) \geq \mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta)$. The lower bound is defined in (15). In this study, the network is, therefore, trained using a loss function defined by maximizing the lower bound in (15). The first term is the expectation of the decoder function under the approximation of the encoder function. This can be thought of as the likelihood of the reconstruction of the data. As a result, the lower bound is maximized by minimizing the reconstruction loss. In this study, the

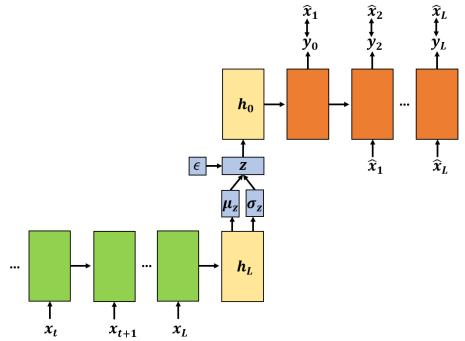


Figure 5: VRAE architecture. The encoder is visualized in green, the final and initial hidden states in yellow, the latent variable in blue and the decoder in orange.

reconstruction loss is evaluated via the mean squared error.

The second term is the negative Kullback-Leibler (KL) divergence between the approximate posterior distribution and the prior. This means that maximizing the lower bound entails minimizing the KL-divergence. This can be thought of as forcing the distribution of the posterior to be close to the distribution of the prior. In effect, this is a regularization term that encourages the latent variables to have Gaussian distributions. In this study, this is evaluated by assuming the prior is distributed as $p_\theta(\mathbf{z}) \sim \mathcal{N}(0, \mathbf{I})$. This regularization term is weighted according to a factor β as seen in (15). For a variational autoencoder, a value of $\beta = 1$ is utilized.

Higgins et al. (2017) introduced the β -variational autoencoder (β -VRAE), often referred to as a disentangled autoencoder. In such an architecture, values of $\beta > 1$ are utilized when training the network. As such, the regulation term is weighted higher. This further encourages the network to learn compact Gaussian distributions. In this manner, the network is able to learn more ordered, i.e. disentangled, representations, since a stronger constraint is imposed on the bottleneck in the latent representation. In this sense, similar data points are encouraged to be closer together in the latent space. As a result, a β -variational recurrent autoencoder may generate more compact and disentangled representations that result in a more effective clustering of AIS trajectories. This architecture is, therefore, chosen for investigation in this study.

$$\begin{aligned} \mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = & \mathbf{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log(p_\theta(\mathbf{x}|\mathbf{z}))] \\ & - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})) \end{aligned} \quad (15)$$

2.4 Trajectory clustering

Trajectory clustering is facilitated via the representations generated in Sec. 2.2 and 2.3. All the trajectories for the region of interest will be run through a forward pass of the encoder for each architecture, resulting in either a hidden state, \mathbf{h} , or latent variable, \mathbf{z} , representation for each trajectory. Clustering can then be

applied to the dataset in either the \mathbf{h} - or \mathbf{z} -space depending on the representation architecture chosen.

In the representation space, standard clustering techniques can be applied. However, due to the unsupervised nature of the problem, the number of clusters is unknown. This is due to the fact that there could be any number of trajectory routes, or groupings of similar trajectories in the data. A technique that can discover the most likely number of clusters is, therefore, necessary. Additionally, there may exist hundreds of trajectory clusters, and a method that can handle such a dataset must, therefore, be utilized. In this study the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm is investigated to facilitate trajectory clustering.

2.4.1 HDBSCAN algorithm

The HDBSCAN algorithm was introduced in Campello et al. (2013). Density-based clustering approaches, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al. 1996), are powerful in that they provide a non-parametric clustering approach that is capable of handling clusters of variable shape, in addition to identifying noise in the data. However, DBSCAN requires hyperparameters that determine the sensitivity of the clustering to noise, and constrict the clusters to having similar densities. HDBSCAN introduces a hierarchical approach that allows for the discovery of clusters of varying density, and is, therefore, more powerful for the case of AIS trajectory clustering.

HDBSCAN begins by finding the core distance of each point d_c , i.e. the distance to the k^{th} -nearest neighbor. The value of k is input as a hyper-parameter. This functions as a local density estimate for the point. The algorithm then calculates a distance metric given by the mutual reachability distance d_m , where the distance between two points \mathbf{x}^i and \mathbf{x}^j is calculated in (16).

Using the mutual reachability distance metric, a minimum spanning tree is constructed, and subsequently converted into a hierarchy of connected components. An additional hyper-parameter is introduced that defines the minimum size of a cluster. Clusters in the hierarchy that do not have a size larger than this value are filtered out. It is then desirable to choose the clusters that have the greatest stability in the hierarchy. This choice, however, must be made under the constraint that once a cluster is selected, no cluster that is a descendant of it may be selected. This results in a clustering scheme where the most stable clusters are found, and all points not belonging to these clusters are labeled as noise. For further details, see Campello et al. (2013).

$$d_m(\mathbf{x}^i, \mathbf{x}^j) = \max(d_c(\mathbf{x}^i), d_c(\mathbf{x}^j), \|\mathbf{x}^i - \mathbf{x}^j\|_2) \quad (16)$$

3 RESULTS AND DISCUSSION

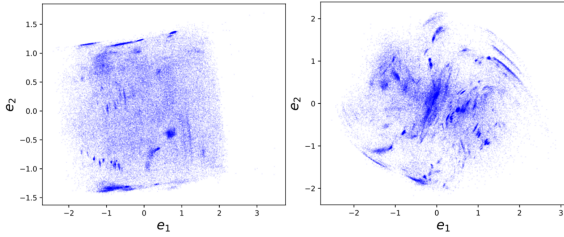
In this section, the power of deep representation learning to facilitate effective AIS trajectory clustering is investigated in a case study of the region surrounding the city of Tromsø, Norway. One year of historical AIS data from January 1st, 2017 to January 1st, 2018 was utilized. This corresponds to 81033 unique trajectories, each of varying length. Zero-padding was utilized on the trajectories to facilitate batch-training, where each trajectory was padded such that it had a length equal to that of the longest trajectory in the batch. Additionally, reversing the order of the input sequence when training recurrent autoencoder architectures has been found to make the optimization easier, as the model can start by looking at low range correlations (Sutskever et al. 2014). Each trajectory, therefore, had its order reversed after padding before being fed to the encoder. The decoder, however, strives to reconstruct the trajectories in the forward direction.

The RNNs utilized in this study had a hidden size of 50 neurons. As a result, the dimensionality of the latent representation for the RAE was equal to the dimensionality of the hidden state, i.e. $\mathbf{h}_t \in \mathbb{R}^{50 \times 1}$. For the β -VRAE, the latent vector was chosen to have a dimensionality of 20, i.e. $\mathbf{z} \in \mathbb{R}^{20 \times 1}$. Furthermore, a value of $\beta = 20$ was utilized for the β -VRAE. A number of iterations using various hyperparameters were conducted, with good results for those presented in this study. PyTorch (Paszke et al. 2019) was utilized to implement the neural networks. All models were trained using the Adam optimizer (Kingma and Ba 2015).

3.1 Deep representation generation

In this section, the deep representations of the historical AIS trajectories are presented. In Fig. 6(a), the final hidden representation, \mathbf{h}_L , representing the latent space of the RAE is illustrated. Each data point represents a unique trajectory in the dataset, facilitated by a forward pass of the encoder of the RAE. As such, the figure illustrates the distribution of the trajectories in the latent space. The latent space for the RAE has 50 dimensions, as the hidden state is 50-dimensional. The top two principle components of the 50 dimensional data are, therefore, visualized in the figure via PCA. The figure indicates that there are groupings of data discovered by the RAE. These should be easily identified when clustering in this space. However, the data do appear to be spread out significantly, which may prove difficult to cluster effectively. Nonetheless, it appears that the RAE is able to discover meaningful representations of the historical AIS data for the selected region in this case.

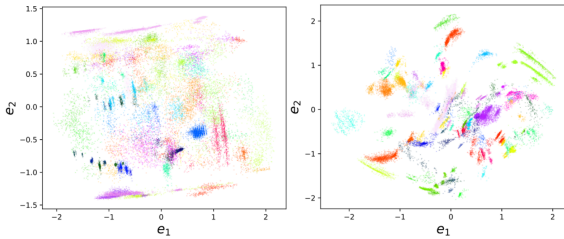
In Fig. 6(b), the latent distribution of the β -VRAE is illustrated. The latent space in this case is 20-dimensional. In order to effectively visualize the data,



(a) RAE.

(b) β -VRAE

Figure 6: Latent distributions, i.e. deep representations. The data are illustrated using the top two principle components of the latent data, e_1 and e_2 .



(a) RAE.

(b) β -VRAE.

Figure 7: Clustering results. The data are illustrated using the top two principle components of the latent data, e_1 and e_2 .

the top two principle components are illustrated via PCA. Comparing the deep representations generated by the β -VRAE in Fig. 6(b) to those from the RAE in Fig. 6(a), it appears that the β -VRAE generated more compact clusters than the RAE. The data are spread out, but appear to be significantly more compact than that in Fig. 6(a). The results indicate, therefore, that the β -VRAE is more effective in generating meaningful representations suitable for clustering, due to the increased grouping of data in the latent space.

3.2 Trajectory clustering

Utilizing the trajectory representations in Sec. 3.1, HDBSCAN was applied to the latent distributions of the RAE and β -VRAE with a minimum cluster size of 50. This was deemed a minimum size to be considered a significant cluster of ship trajectories. The implementation in McInnes et al. (2017) was utilized to facilitate the clustering. The results for the RAE and β -VRAE are illustrated in Fig. 7(a) and 7(b) respectively. The reader should note that multiple clusters appear to have similar colors, but are in fact separate. For the RAE, 103 clusters were discovered, and 111 for the β -VRAE. HDBSCAN appears to have effectively clustered the data, where clusters of varying density and shape were discovered. Furthermore, these architectures can be utilized to support anomaly detection, either via the identified noise data from HDBSCAN, or by further analysis of individual clus-

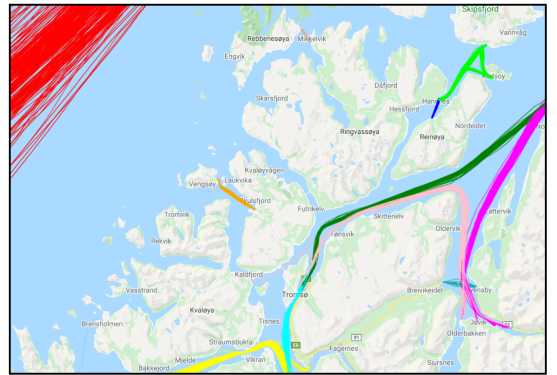


Figure 8: Subset of clustered trajectories for the region of Tromsø, Norway. Map courtesy of Google Maps (2020).

ters. As indicated by the results in Sec. 3.1, however, it appears that representations generated using the β -VRAE resulted in a more effective clustering regime. The RAE generated clusters that were much less dense and defined, whilst the β -VRAE generated more compact and well defined clusters. As a result, the results indicate that a β -VRAE has superior performance with respect to generating deep representations to facilitate clustering.

Clustering using both the RAE and β -VRAE resulted in over 100 different clusters. Therefore, it is very difficult to visualize the clustered trajectories on a map. A subset consisting of 11 clusters from the β -VRAE results are presented in Fig. 8 to illustrate the clustering performance. The reader should note that the colors do not match those for the clustering results in Fig. 7. Given the unsupervised nature of the problem, the clustering must be validated by a visual inspection of the raw trajectories on a map. The results indicate that the deep representations in conjunction with the HDBSCAN algorithm resulted in a successful clustering of the trajectories, with the β -VRAE having the best performance.

4 CONCLUSION

In this study, it has been shown that deep representation learning can provide powerful representations of historical AIS trajectories to facilitate vessel trajectory clustering. By utilizing recurrent autoencoder architectures, variable length trajectories can be encoded to a fixed size vector. A recurrent autoencoder and β -variational recurrent autoencoder were compared. Both architectures were found effective in generating groupings of similar trajectories. However, the β -variational recurrent autoencoder provided more compact representations. Using the region of Tromsø, Norway as a test case, the Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm was applied to the deep representations, providing a successful clustering of the data. Based on a visual inspection of the clusters plotted on a map, the trajectory clusters appear to be meaningful.

The method outlined in this study can be expanded to include more complex architectures such as stacked- and bi-directional recurrent autoencoders that likely will be able to capture more information in their deep representations. Nonetheless, the less complex models described in this study appear to provide a successful clustering of the trajectories for the tested region. Further work will investigate more complex deep learning architectures that can be used in conjunction with trajectory prediction algorithms for collision avoidance purposes.

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, Industry and Fisheries. The authors would also like to thank the Norwegian Coastal Administration for providing access to their AIS database.

REFERENCES

- Aarsæther, K. G. & T. Moan (2009). Estimating Navigation Patterns from AIS. *Journal of Navigation* 62(04), 587–607.
- Bengio, Y., P. Simard, & P. Frasconi (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 5(2), 157–166.
- Bourlard, H. & Y. Kamp (1988). Auto-association by multi-layer perceptrons and singular value decomposition. *Biological Cybernetics* 59(4-5), 291–294.
- Campello, R. J. G. B., D. Moulavi, & J. Sander (2013). Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu (Eds.), *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg, pp. 160–172. Springer Berlin Heidelberg.
- Cho, K., B. van Merriënboer, D. Bahdanau, & Y. Bengio (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches.
- Chung, J., C. Gulcehre, K. Cho, & Y. Bengio (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS'2014 Deep Learning workshop*.
- Ester, M., H.-P. Kriegel, J. Sander, & X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD96, pp. 226231. AAAI Press.
- Fabius, O. & J. R. van Amersfoort (2015). Variational Recurrent Auto-Encoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Goodfellow, I., Y. Bengio, & A. Courville (2016). *Deep Learning*. MIT Press.
- Google Maps (2020). Map of Tromsø, Norway.
- Higgins, I., L. Matthey, A. Pal, C. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, & A. Lerchner (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Hinton, G. E. & R. R. Salakhutdinov (2006). Reducing the dimensionality of data with neural networks. *Science* 313(5786), 504–507.
- Hochreiter, S. & J. Schmidhuber (1997). Long Short-Term Memory. *Neural Computation* 9(8), 1735–1780.
- Kingma, D. P. & J. L. Ba (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Kingma, D. P. & M. Welling (2014). Auto-Encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- McInnes, L., J. Healy, & S. Astels (2017). hdbSCAN: Hierarchical density based clustering. *The Journal of Open Source Software* 2(11), 205.
- Murray, B. & L. P. Perera (2020). A Dual Linear Autoencoder Approach for Vessel Trajectory Prediction Using Historical AIS Data. *Ocean Engineering*.
- Nguyen, D., R. Vadaine, G. Hajduch, R. Garelo, & R. Fablet (2018). A multi-task deep learning architecture for maritime surveillance using ais data streams. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 331–340.
- Pallotta, G., M. Vespe, & K. Bryan (2013). Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. *Entropy* 15(12), 2218–2245.
- Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, & S. Chintala (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems* 32, pp. 8024–8035. Curran Associates, Inc.
- Rezende, D. J., S. Mohamed, & D. Wierstra (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML14*.
- Rumelhart, D. E., G. E. Hinton, & R. J. Williams (1986). Learning representations by back-propagating errors. *Nature* 323(6088), 533–536.
- Spinner, T., J. Krner, J. Grtler, & O. Deussen (2018). Towards an interpretable latent space : an intuitive comparison of autoencoders with variational autoencoders. In *Proceedings of the Workshop on Visualization for AI Explainability 2018 (VISxAI)*.
- Srivastava, N., E. Mansimov, & R. Salakhutdinov (2015, feb). Unsupervised Learning of Video Representations using LSTMs. *32nd International Conference on Machine Learning, ICML 2015 1*, 843–852.
- Sutskever, I., O. Vinyals, & Q. V. Le (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112.
- Tu, E., G. Zhang, L. Rachmawati, E. Rajabally, & G.-B. Huang (2017). Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology. *IEEE Transactions on Intelligent Transportation Systems*, 1–24.
- Yao, D., C. Zhang, Z. Zhu, J. Huang, & J. Bi (2017). Trajectory clustering via deep representation learning. In *Proceedings of the International Joint Conference on Neural Networks*, Volume 2017-May, pp. 3880–3887. Institute of Electrical and Electronics Engineers Inc.
- Zhang, S.-k., G.-y. Shi, Z.-j. Liu, Z.-w. Zhao, & Z.-l. Wu (2018). Data-driven based automatic maritime routing from massive AIS trajectories in the face of disparity. *Ocean Engineering* 155, 240–250.
- Zhou, Y., W. Daamen, T. Vellinga, & S. P. Hoogendoorn (2019). Ship classification based on ship behavior clustering from AIS data. *Ocean Engineering*, 176–187.

Paper V

An AIS-Based Deep Learning Framework for Regional Ship Behavior Prediction

Brian Murray and Lokukaluge Prasad Perera (2021)

Submitted and in First Revision for Publication in *Journal of Reliability Engineering and System Safety. Special Issue on Safety of Maritime Transportation Systems.*

An AIS-Based Deep Learning Framework for Regional Ship Behavior Prediction

Brian Murray*, Lokukaluge Prasad Perera

UiT The Arctic University of Norway, Tromsø, Norway

ARTICLE INFO

Keywords:

Maritime Safety
Maritime Situation Awareness
Ship Navigation
Trajectory Prediction
Collision Avoidance
Deep Learning
AIS

ABSTRACT

This study presents a deep learning framework to support regional ship behavior prediction using historical AIS data. The framework is meant to aid in proactive collision avoidance, in order to enhance the safety of maritime transportation systems. In this study, it is suggested to decompose the historical ship behavior in a given geographical region into clusters. Each cluster will contain trajectories with similar behavior characteristics. For each unique cluster, the method generates a local model to describe the local behavior in the cluster. In this manner, higher fidelity predictions can be facilitated compared to training a model on all available historical behavior. The study suggests to cluster historical trajectories using a variational recurrent autoencoder and the Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm. The past behavior of a selected vessel is then classified to the most likely clusters of behavior based on the softmax distribution. Each local model consists of a sequence-to-sequence model with attention. When utilizing the deep learning framework, a user inputs the past trajectory of a selected vessel, and the framework outputs the most likely future trajectories. The model was evaluated using a geographical region as a test case, with successful results.

Nomenclature

b	Bias Vector
c	Class Vector
e	Principle Component
<i>f</i>	Arbitrary Function
h	Hidden State
<i>J</i>	Loss Function
<i>L</i>	Sequence Length
n	New Candidate Vector
<i>N</i>	Number of Layers
<i>p</i>	Probability
p	Position Vector
<i>q</i>	Approximate Encoder
r	Reset Gate
s	Static Data
u	Update Gate
v	Speed over Ground
v	Input to Softmax Layer
W	Weight Matrix
x	Input Sequence
y	Target Sequence
z	Latent Representation Vector
β	Weighting Hyperparameter
χ	Course over Ground Vector
μ	Mean Vector
σ	Standard Deviation
Θ	Model Parameters
<i>Subscripts</i>	
cat	Categorical
cont	Continuous
<i>h</i>	Hidden
<i>i</i>	Class Number

Subscripts

<i>n</i>	New Candidate
<i>r</i>	Reset
<i>t</i>	State
<i>u</i>	Update
<i>x</i>	Input
<i>z</i>	Latent Representation
ϕ	Encoder Parameters
θ	Decoder Parameters

Superscripts


$\hat{}$	Estimated Parameter / State
<i>l</i>	Layer

Acronyms

AIS	Automatic Identification System
GRU	Gated Recurrent Unit
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
KL	Kullback-Leibler
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
VRAE	Variational Recurrent Autoencoder

1. Introduction

Effective maritime traffic monitoring is essential for maintaining the integrity of maritime transportation systems. The safety of human life, as well as that of material assets, and the ocean environment, depend on conducting safe maritime operations. Maritime situation awareness can be argued to be one of the most essential elements with regards to maintaining the safety of such systems. Situation awareness is defined as being aware of what is happening around oneself, and understanding the implications of the current situation now, as well as in the future (Endsley et al., 2003). All navigators must have an adequate degree of situation awareness

 brian.murray@uit.no (B. Murray)
ORCID(s):

to effectively conduct operations at sea. In this context, the primary challenge relates to detecting obstacles and predicting close-range encounter situations. As such, effective collision avoidance can be viewed as a key component of safe maritime transportation systems.

Navigators rely on visual observation, as well as any navigational tools they have available to them, to maintain an adequate degree of situation awareness. Such tools include radar, conning, ECDIS (Electronic Navigation Chart Display and Information System) and AIS (Automatic Identification System). With respect to collision avoidance, navigators rely heavily on radar systems facilitated by ARPA (Automatic Radar Plotting Aid) in addition to the ECDIS. The best navigational tools should be available to navigators to support the navigator in identifying high risk situations (Perera and Guedes Soares, 2015), such that they can conduct effective collision avoidance maneuvers that adhere to the COLREGS (Perera et al., 2010).

Generally, a linear constant velocity model is utilized to evaluate potential close-encounter situations. In this manner, the future position of a vessel is predicted using constant speed and course over ground values. This method is reliable, and provides the basis for many commercial systems for predictive traffic surveillance (Xiao et al., 2020). However, they are inherently constrained by their linearity, and will have degraded performance when predicting complex behavior. More advanced techniques, e.g. Perera et al. (2012); Perera (2017), where extended Kalman filters were utilized to estimate more complex ship behavior, can aid in predicting more complex ship behavior. However, such techniques will not be useful for prediction horizons greater than a few minutes.

Perera and Murray (2019) suggested to introduce an advanced ship predictor to aid maritime situation awareness. The predictor is comprised of a local and global predictor to overcome such issues. On a local scale, such techniques can be used to predict short-term ship behavior (order 0-5 minutes). A global predictor is used to predict more long-term behavior (order 5-30 minutes). The goal of such global predictions is to prevent close-encounter situations from arising. By predicting the future trajectory of vessels accurately, the future collision risk between two neighboring vessels can be computed. In this manner, the risk of future close-encounter situations can be predicted, and appropriate collision avoidance actions implemented (Daranda, 2016). Such global behavior may, however, be complex, and will require more advanced techniques to effectively predict.

Developments within maritime traffic monitoring systems can assist in providing situation awareness to navigators, such that proactive collision avoidance maneuvers can be conducted. Vessel Traffic Systems (VTS) collect traffic data from a variety of sources, including AIS, shore-based radar, Long-Range Identification and Tracking, as well as Synthetic Aperture Radar (SAR) satellite imagery to support maritime traffic safety. The data from such real-time observations are used by VTS operators to support proactive traffic management (Xiao et al., 2020). The ubiquity of

data relating to maritime traffic opens up for opportunities to take advantage of recent developments in machine learning and artificial intelligence.

1.1. Historical AIS Data

Historical AIS data provide insight into the historical behavior of ships in given regions, which can be used for maritime traffic data mining and forecasting techniques. Research into utilizing these data to support maritime transportation systems has been the topic of much research recently, with a review of various applications found in (Tu et al., 2017). For instance, Goerlandt and Kujala (2011) utilized AIS data to simulate maritime traffic and assess the probability of ship collisions. Silveira et al. (2013) evaluated the ship collision risk off the coast of Portugal, providing a statistical analysis of the traffic separation schemes and evaluate collision risk. Rong et al. (2020) also utilized AIS data to characterize maritime traffic and detect anomalies using data mining. A review of methods to assess waterway risk based on AIS data can also be found in Du et al. (2020).

1.1.1. AIS-Based Ship Behavior Prediction

Ristic et al. (2008) was one of the first to investigate using AIS data for trajectory prediction. The study used a particle filter for ship behavior prediction based on AIS data. The uncertainty of the prediction, however, renders the method of limited use with respect to collision avoidance purposes. A number of studies have also addressed clustering historical AIS trajectories, classifying a vessel to a given cluster and conducting a prediction. Pallotta et al. (2013) introduced the TREAD (Traffic Route Extraction and Anomaly Detection) method to cluster historical trajectories into routes, and classify a partial trajectory to one of these routes. Pallotta et al. (2014) expanded this work to predict vessel positions for a cluster discovered by TREAD via an Ornstein Uhlenbeck stochastic process. Mazzarella et al. (2015) also applied a bayesian network approach using a particle filter for trajectory prediction. These methods, however, are useful for predictions in the order of hours, and as such of greater benefit for general maritime traffic forecasting, than for collision avoidance purposes.

Other methods include Hexeberg et al. (2017) which introduced a single point neighbor search method to predict trajectories using historical AIS data. The method, however, does not handle branching waterways, and the accuracy of the method is limited. Dalsnes et al. (2018) expanded this approach to provide multiple predictions using a prediction tree. The resultant predictions are then clustered using a Gaussian mixture model. Both these methods, however, do not utilize trajectory clustering prior to conducting a prediction. Predictions are based on the neighborhood of a predicted state, which may include data points that belong to other clusters of ship behavior, inherently degrading the performance.

Rong et al. (2019) presented a probabilistic approach to ship behavior prediction using a Gaussian process model. This method had successful results for the investigated region off the coast of Portugal. However, this region did not

contain complex traffic situations. Therefore, the outcome of the same approach to more complex traffic regions is inconclusive. Based on a clustering of locally extracted trajectories, Murray and Perera (2020a) classified a selected vessel to one of the clusters, and predicted the future trajectory using a dual linear autoencoder approach. This approach had successful results, but was computationally expensive with respect to extracting trajectories. This may degrade the results in certain situations with respect to collision avoidance purposes.

1.1.2. Deep Learning-Based Approaches

Deep learning (Goodfellow et al., 2016) has been the center of technological innovation in recent years. With state of the art performance in image and speech recognition (Goodfellow et al., 2016), the methods have slowly begun to gain the attention of other domains. Within the maritime domain, however, there is still limited research on adopting deep learning techniques.

AIS data are an ideal dataset to apply deep learning techniques in the maritime domain. Zhang et al. (2020), for instance, applied a convolutional neural network to classify regional ship collision risk levels. Nguyen et al. (2018) developed a multi-task deep learning architecture for maritime surveillance based on a variational recurrent neural network. The framework can be utilized for multiple purposes including trajectory prediction. However, the method applied a 4-hot encoding to the data that reduces the resolution of the predictions, degrading the performance with respect to collision avoidance purposes.

Yao et al. (2017) investigated clustering AIS trajectories using deep representation learning, where the results indicated that the deep learning approach outperformed non-deep learning based approaches. Murray and Perera (2020b) expanded this work, where it was found that a variational recurrent autoencoder architecture provided better representations for trajectory clustering. These methods, however, do not provide a method to predict the future trajectory of a selected vessel. Forti et al. (2020) utilized a recurrent neural network to predict trajectories using a sequence-to-sequence model. The results are promising, but have only been tested on a dataset of limited complexity. If applied to an entire region of historical data, the performance will likely be degraded.

1.2. Contribution

In this study, it is suggested to utilize historical AIS data to predict ship behavior on a global scale, with the purpose of aiding in proactive collision avoidance. It is, therefore, assumed that future ship behavior can be predicted based on the historical behavior of other vessels in a given geographical region. If successful, such methods can aid in providing situation awareness to navigators, VTS centers, as well as future autonomous vessels.

The study presents a deep learning framework for regional ship prediction. Given the past trajectory of a selected vessel, the framework predicts its future trajectory. To facilitate this, the data for a specific geographical region are

used to generate a prediction model for ship behavior within this region. Similar methods train neural networks on all the available AIS data. However, for the purpose of aiding in collision avoidance, trajectory predictions should be as accurate as possible. As a result, this study suggests decomposing the historical ship behavior into local models.

To create these local models, it is suggested to cluster historical ship behavior using a variational recurrent autoencoder, as outlined in Murray and Perera (2020b). This approach is expanded to add more complexity to the model, resulting in improved clustering performance. The method is able to discover clusters of ship behavior, such that local models can be trained for each individual cluster. Such local models should have enhanced performance, as they are trained on specific ship behavior. In contrast, training on all available data will result in models that must capture a much larger degree of variation, inherently degrading their performance due to the increased complexity of the underlying data.

The method further suggests to classify a trajectory segment to a given cluster using a deep learning architecture. The method outputs a distribution over possible clusters of behavior the trajectory may belong to, such that multiple predictions can be made. It is highly likely that the trajectory belongs to one of these clusters, and as a result, one of the trajectory predictions should be accurate.

The local models are trained on the data in each unique cluster of historical behavior. In this study, a sequence-to-sequence model using an attention mechanism is suggested to function as the local model for each cluster. This architecture has not previously been addressed for AIS-based trajectory in the literature to the best of the authors knowledge. Such attention mechanisms provide the basis for state of the art translation architectures, improving the performance significantly compared to conventional sequence-to-sequence models. Furthermore, sequence-to-sequence models should have enhanced performance compared to standard recurrent neural networks, as they predict an entire sequence based on an entire input sequence. As such the error for the entire future sequence is used to optimize the network, and not just error for one time step at a time.

The overall framework outlined in this study provides a novel contribution to conduct efficient trajectory predictions. Using pre-trained networks for a given geographical region, the most probable future trajectories can be output to a user in under a second.

2. Methodology

In this section, the proposed methodology of the deep learning framework is outlined. The framework is designed such that it can be applied to any geographical region. The objective of the framework is to support ship behavior prediction. It is assumed that the respective vessels observed in a given geographical region may have similar behavior to that of other vessels in the past. By developing a framework to model the historical behavior of the respective ships for a given region, it may be possible to predict the future behav-

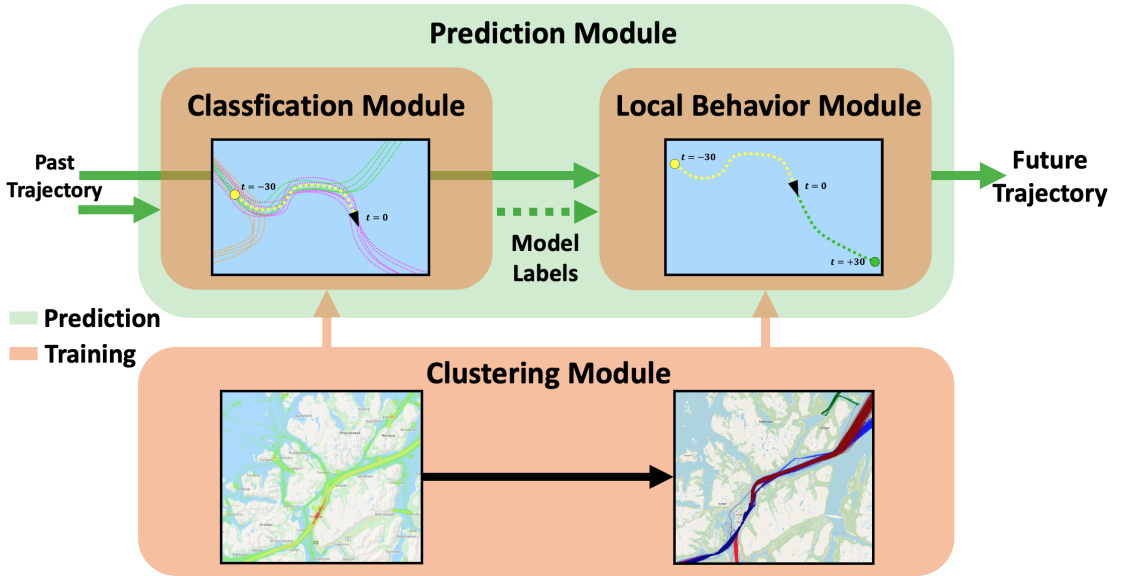


Figure 1: Overview of deep learning framework.

ior of a selected vessel. This is achieved through the use of historical AIS data.

An overview of the framework is illustrated in Fig. 1. Overall, the framework can be viewed as being conducted in two phases. The first is the training of the modules, where the trained models are illustrated in orange in Fig. 1. The second phase is the prediction phase, and uses the pre-trained networks. This phase is illustrated in Fig. 1 in green.

The clustering module is trained first using all available historical AIS data for the selected geographical region. The goal of the clustering module is to discover clusters of historical ship behavior. These clusters contain historical AIS trajectories that have similar behavior. Fig. 1 depicts the clustering module in orange. The left figure in the module presents a heat map of the available historical AIS data for a specified region, and the right a subset of clusters of ship behavior. These clusters correspond to groupings of similar historical ship behavior. Such clusters may, for instance, involve alternate routes, or speed profiles along routes.

The purpose of the prediction module is to predict the future trajectory of a selected vessel, given its observed past behavior. It is assumed in this study that the past 30 minutes of AIS data are available. The input to the prediction module, as illustrated in Fig. 1 corresponds, therefore, to the past 30 minute behavior of a selected vessel. However, the architecture can be trained based on any input trajectory length.

The prediction module consists of two sub-modules, the classification module and local behavior module. In the classification module, the input trajectory is matched to one of the behavior clusters discovered in the clustering module.

The input trajectory in this study is the past 30 minute behavior of the selected vessel. The classified cluster label is then input to the local behavior module, which selects the pre-trained model that corresponds to that cluster of behavior. This model is then used to predict the future 30 minute behavior of the selected vessel. The classification module also outputs multiple possible clusters the trajectory may belong to, with a probability associated with each cluster. In this manner, multiple trajectories can be predicted based on the local models for the classified behavior clusters.

2.1. Preprocessing

Prior to training the neural networks involved in this study, preprocessing of the AIS data must be conducted. The first step is to generate complete trajectories from the unprocessed AIS data. This is conducted by extracting trajectory segments, where the time between consecutive points exceeds some parameter. In this study, trajectories are defined where any two points are more than 30 minutes apart. Furthermore, each individual trajectory is interpolated at one minute intervals to facilitate higher density data, as well as provide a common foundation for training the network. As such, each vessel state, \mathbf{x}_t , will be one minute apart. Each vessel state is defined in (1), and contains the positional data defined in UTM coordinates $\mathbf{p} = [p_1, p_2]$, as well as the speed over ground v and course over ground, χ decomposed into the UTM coordinate directions, $\chi = [\chi_1 \chi_2]$.

$$\mathbf{x}_t = [p_1, p_2, v, \chi_1, \chi_2] \quad (1)$$

However, the parameters in the vessel states vary significantly in magnitude. As such, all states are scaled across

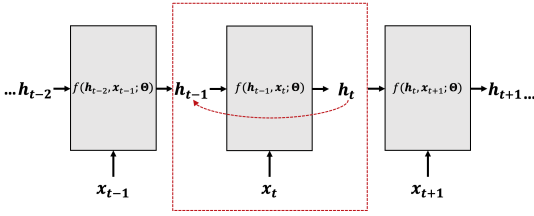


Figure 2: RNN. Illustration adapted from Murray and Perera (2020b)

each parameter for all extracted trajectories from the region of interest. In this case, the values are scaled between $[-1, 1]$ given that the data is more optimal for the recurrent neural networks that make use of the tanh function.

All trajectories are present in the input data, i.e. no anomalous trajectories have been removed from the dataset. This is due to the ability of the clustering module to identify such trajectories, and remove them before further processing. This is addressed in Sec. 2.3.

2.2. Recurrent Neural Networks

Recurrent neural networks (RNNs) (Rumelhart et al., 1986) are designed to handle sequence data. The general RNN architecture is visualized in Fig. 2. As historical AIS trajectories are multivariate time series, RNNs are chosen to serve as the main deep learning architecture in the framework utilized in this study. RNNs are capable of handling time series of variable length, and can be combined with other architectures to achieve various goals. RNNs are ideal for time series data in that they incorporate a sense of memory into the network. Given a time series $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_L\}$ of length L , a recurrent neural network processes the input state \mathbf{x}_t at a given state, t , sequentially. In addition, information about the time series before state t is processed through previous hidden state \mathbf{h}_{t-1} . The network then outputs the current hidden state \mathbf{h}_t that incorporates relevant information from \mathbf{x}_t and \mathbf{h}_{t-1} in (2).

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t; \Theta) \quad (2)$$

Each operation can be thought of as applying (2) in a RNN cell. The same operation repeats for all states, and is in this sense recurrent. A recurrent neural network can be thought of as an unfolded computational graph, where each operation, i.e. each cell, applies (2). In this manner the parameters are shared between all operations. The recurrence is visualized within the red box in Fig. 2, indicating that each cell is fed the previous cells output along with the current input. The architecture is in this sense causal, where the current output depends on all the past time steps. It is evident that such an architecture is applicable to ship trajectories, in that the future behavior should be dependent on the past behavior.

2.2.1. Gated Recurrent Unit

The original RNN architecture is often referred to as the vanilla RNN. When training this architecture, the network

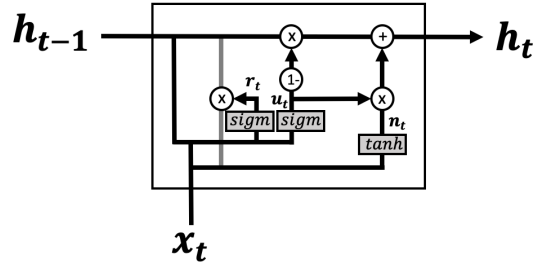


Figure 3: GRU cell. Illustration adapted from Murray and Perera (2020b).

struggles to learn long-term dependencies. This is due to vanishing gradients during backpropagation of the network (Bengio et al., 1994). The long-term memory of such networks is, therefore, poor, and can degrade their performance when long-term dependencies in the data exist. The Gated Recurrent Unit (GRU) (Cho et al., 2014; Chung et al., 2014) is an recurrent architecture that introduces the concept of gates to reduce the effect of vanishing gradients. Fig. 3 illustrates the architecture of the GRU cell. Other gated architectures include the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). The GRU, however, reduces the number of model parameters compared to the LSTM, thereby reducing training time.

In the GRU, the reset gate (3) and the update gate (4) regulate how much information should be passed on to the next cell, i.e. how much should be remembered.

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{b}_{xr} + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r) \quad (3)$$

$$\mathbf{u}_t = \sigma(\mathbf{W}_{xu}\mathbf{x}_t + \mathbf{b}_{xu} + \mathbf{W}_{hu}\mathbf{h}_{t-1} + \mathbf{b}_{hu}) \quad (4)$$

When training the network, the weights in each weight matrix, \mathbf{W} , and bias vector, \mathbf{b} , are updated. A sigmoid activation function outputs values between zero and one. These values are then multiplied with the data to be passed on using the Hadamard product, thereby either restricting or allowing the flow of information. A new candidate vector is calculated in (5) using a hyperbolic tangent activation function, \tanh . The current hidden state is then calculated in (6) and passed on to the next state.

$$\mathbf{n}_t = \tanh(\mathbf{W}_{xn}\mathbf{x}_t + \mathbf{b}_{xn} + \mathbf{r}_t \odot (\mathbf{W}_{hn}\mathbf{h}_{t-1} + \mathbf{b}_{hn})) \quad (5)$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \mathbf{n}_t \quad (6)$$

The internal functionality of the GRU cell can, therefore, be described in equations (3)-(6), where the general form of the RNN cell in (2) is equivalent to (6). For further details see Cho et al. (2014); Chung et al. (2014).

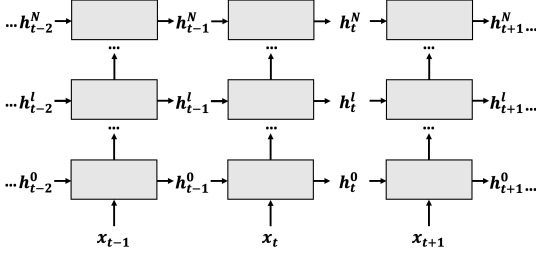


Figure 4: Stacked RNN with N layers.

2.2.2. Bidirectional RNNs

Standard RNNs conduct calculations in the forward direction, i.e. from the past to the future. Bidirectional RNNs (Schuster and Paliwal, 1997), however, provide an architecture where the calculations are conducted in both the forward and backward directions concurrently. In this manner, future events can be thought to affect past events. In the case of ship trajectories, this may not be as intuitive. However, the argument can be made that choices made by a navigator may depend on future choices, e.g. speed changes dependent on a future course alteration, route choice, etc. As such, a bidirectional RNN will incorporate more information about the navigational patterns of past ships in a historical AIS data set.

2.2.3. Stacked RNNs

Deep neural networks, i.e. with multiple layers, have been shown to have superior performance to more shallow networks. The same can be said for RNNs, as it was shown in Graves et al. (2013) that increasing depth to RNNs enhanced their performance. Such RNNs are often referred to as stacked RNNs. The stacked architecture merely implies that there are multiple RNNs that feed into each other as illustrated in Fig. 4. The figure illustrates a network of N layers with 0 being the initial layer and arbitrary layer l between.

2.3. Clustering Module

In the clustering module, clusters of ship behavior are discovered. This is achieved through an unsupervised learning technique known as clustering, where the underlying groupings in the data are discovered. The groupings in the case of this study correspond to sets of trajectories with similar behavior. Discovering such groupings, however, can be challenging. Standard clustering techniques require representations of the data to be vectors of equal size. A clustering algorithm will then group the vectors based on some similarity, i.e. distance, measure. Historical AIS trajectories, however, consist of multivariate time series with variable length. As a result, they can not be clustered using standard techniques. It is, therefore, of interest to develop a framework to generate fixed size representations of the trajectories, such that standard clustering techniques can then be applied to the representations.

Murray and Perera (2020b) suggested to utilize a deep representation learning-based approach to facilitate trajectory representation generation for subsequent clustering. The study argues that RNNs are ideal for such a task, as they are designed to generate representations of multivariate sequences via their hidden states. The study compares utilizing a recurrent autoencoder and β -variational recurrent autoencoder (β -VRAE) (Higgins et al., 2017) to learn good representations of the data. It was found that the β -VRAE provided more compact groupings, resulting in a more effective clustering scheme. The Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm (Campello et al., 2013) was utilized to cluster the trajectory representations with good results.

The method is expanded in this study, where a bidirectional stacked VRAE architecture is utilized to generate representations of the data, which are subsequently clustered using HDBSCAN. The details of the architecture of the clustering module are presented in the following sections.

2.3.1. Variational Recurrent Autoencoder

The goal of the VRAE is to generate meaningful representations of the historical AIS trajectories, such that they can be effectively clustered. A common method to generate meaningful representations is the autoencoder. An autoencoder is comprised of two parts, an encoder and a decoder. The encoder encodes the data to a latent representation, and the decoder subsequently attempts to reconstruct the data from this latent representation.

RNNs inherently provide a compression of the data, where feeding a sequence into an RNN, the network outputs a final hidden state \mathbf{h}_L that represents the entire input sequence. By training an encoder RNN that encodes the historical ship trajectories to a hidden state, \mathbf{h}_L , a decoder can be trained to reconstruct the input trajectory from \mathbf{h}_L . Such an architecture is known as a recurrent autoencoder (Srivastava et al., 2015). This is in essence an application of sequence-to-sequence models (Sutskever et al., 2014) that provide the basis for the state-of-the-art in natural language processing tasks e.g. translation (Cho et al., 2014).

The VRAE is an extension of the recurrent autoencoder that utilizes a variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014). The VAE introduces a probabilistic approach to the autoencoder, where it is assumed that data are generated by a random process from a continuous latent variable denoted \mathbf{z} . An approximate probabilistic encoder, $q_\phi(\mathbf{z}|\mathbf{x})$, produces a distribution over the latent variable, \mathbf{z} , and a decoder $p_\theta(\mathbf{x}|\mathbf{z})$ reconstructs \mathbf{x} from \mathbf{z} . It is, furthermore, assumed that $q_\phi(\mathbf{z}|\mathbf{x})$ is a multivariate Gaussian with a diagonal covariance in (7).

$$q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2 \mathbf{I}) \quad (7)$$

Both the encoder, $q_\phi(\mathbf{z}|\mathbf{x})$, and decoder, $p_\theta(\mathbf{x}|\mathbf{z})$, are approximated by neural networks. The advantage of using a VAE compared to a standard autoencoder, is that it encourages the latent variables to become normally distributed. Murray and Perera (2020b) argued that this limits the chaos in the latent

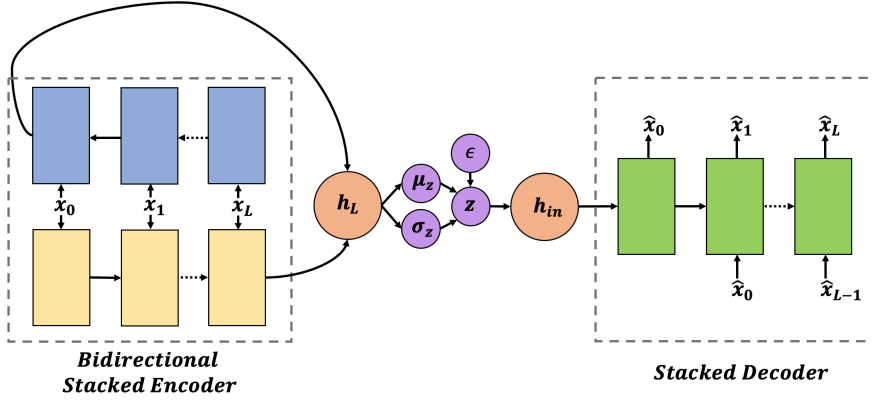


Figure 5: VRAE with a bidirectional stacked encoder, and stacked decoder. The forward encoder is illustrated in yellow, and the backward encoder in blue. The decoder is illustrated in green. All RNNs are stacked.

space, and forces the latent representations of the data to be more compact, thereby providing better representations for a clustering algorithm.

Fabius and van Amersfoort (2015) extended the VAE to introduce a recurrent architecture in the variational recurrent autoencoder (VRAE). Here the encoder and decoder are comprised of RNNs. The overview of the architecture in this study is presented in Fig. 5. To the left in the figure is the encoder. The encoder is bidirectional, where the forward encoder is illustrated in yellow, and the backward encoder illustrated in blue. Both encoders are GRUs. Integrating a bidirectional architecture, more information can be encoded in the latent space. Furthermore, the bidirectional encoder is stacked, providing increased depth to the network. This allows it to learn more complex relationships in the data. The output of the forward and backward encoders are concatenated to comprise the final hidden state, \mathbf{h}_L , of the encoder. This is visualized in orange in Fig. 5. The mean and standard deviation of the normal distribution in (7) are estimated via linear layers in (8) and (9).

$$\boldsymbol{\mu}_z = \mathbf{W}_\mu \mathbf{h}_L + \mathbf{b}_\mu \quad (8)$$

$$\boldsymbol{\sigma}_z = \mathbf{W}_\sigma \mathbf{h}_L + \mathbf{b}_\sigma \quad (9)$$

Using the re-parametrization trick to allow for backpropagation, the latent variable is estimated in (10), where ϵ is sampled from a normal distribution according to $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. This is illustrated in purple in Fig. 5.

$$\mathbf{z} = \boldsymbol{\mu}_z + \boldsymbol{\sigma}_z \odot \epsilon \quad (10)$$

The decoder, illustrated in green in Fig. 5, takes the initial hidden state as input. This is calculated in (11).

$$\mathbf{h}_{in} = \tanh(\mathbf{W}_{zh} \mathbf{z} + \mathbf{b}_{zh}) \quad (11)$$

It then reconstructs the input sequence sequentially, where the next state is estimated according to (12).

$$\hat{\mathbf{x}}_{t+1} = \mathbf{W}_{h\hat{x}} \mathbf{h}_t + \mathbf{b}_{h\hat{x}} \quad (12)$$

Each predicted state is fed into the following cell to predict the next. The basis for the entire prediction is the input from \mathbf{h}_{in} . Therefore, all the information contained in the sequence must be stored in the latent vector \mathbf{z} . Training this encoder-decoder architecture forces the network to learn a meaningful representation of the data in the latent space.

The network is optimized by maximizing a variational lower bound on the log-likelihood (13).

$$J(\theta, \phi; \mathbf{x}, \mathbf{z}) = \mathbf{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log(p_\theta(\mathbf{x}|\mathbf{z}))] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})) \quad (13)$$

The first term in (13) can be viewed as the reconstruction loss, and is evaluated in this study using the mean squared error. The second term is the Kullback-Leibler (KL) divergence between the approximate posterior, $q_\phi(\mathbf{z}|\mathbf{x})$ (i.e. encoder), and the prior $p_\theta(\mathbf{z})$. In this study it is assumed that the prior is normally distributed according to $p_\theta(\mathbf{z}) \sim \mathcal{N}(0, \mathbf{I})$. Therefore, maximizing this second term implies minimizing the KL-divergence. By applying this constraint to the latent space, this term tries to enforce compact groupings of data. Therefore, similar ship trajectories should be encouraged to be closer together in the latent space. As such, a clustering algorithm should be more successful in discovering clusters of trajectories using such an architecture. For further details on representation learning for trajectory clustering, please see Murray and Perera (2020b) as well as Kingma and Welling (2014); Rezende et al. (2014); Fabius and van Amersfoort (2015) for further details on VAEs.

2.3.2. Hierarchical Density-Based Spatial Clustering of Applications with Noise

The Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm (Campello et al., 2013) is utilized to cluster the latent representations (i.e. \mathbf{z}). The goal is to identify class labels that can be utilized in the classification module, where each class relates to a cluster of ship behavior. HDBSCAN is a non-parametric clustering

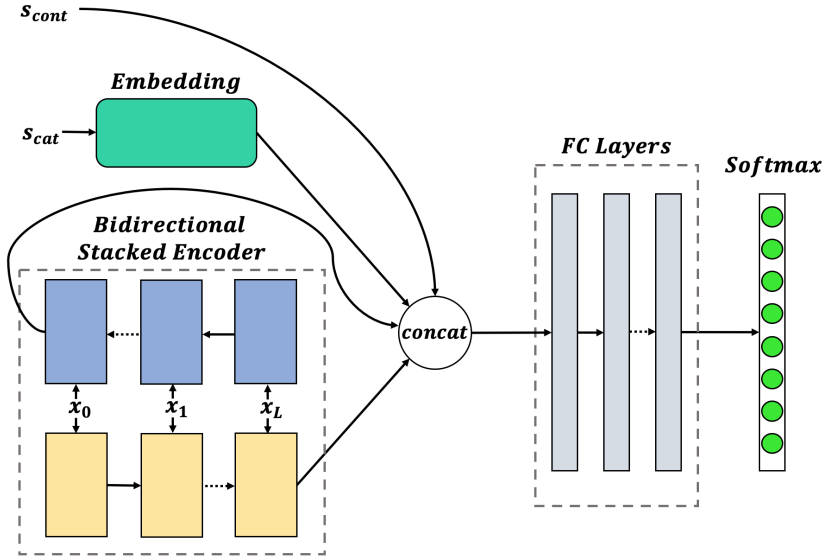


Figure 6: Classifier with a bidirectional stacked encoder for dynamic data. The embedding block embeds categorical static data. The encoded dynamic data is concatenated with the embedded categorical static data and continuous static data. Fully connected (FC) layers predict the class via a softmax output layer.

approach that can identify clusters of varying density and shape, and was argued in Murray and Perera (2020b) to be powerful in clustering ship trajectory representations generated by a VRAE.

HDBSCAN extends the Density-Based Spatial Clustering of Applications with Noise algorithm by adapting it to a hierarchical clustering scheme. The algorithm defines core distances for each point as the distance to the k^{th} nearest neighbor. These distances function as local density estimates, and provide the basis for a mutual reachability metric between two points. This metric then provides the basis for a minimum spanning tree and hierarchy. The tree is then pruned using the minimum cluster size, where any clusters below a given threshold are filtered out. The algorithm then discovers the most stable clusters in the hierarchy. Furthermore, HDBSCAN provides the capability to discover noise in the data, where any data points that do not belong to the clusters are labeled as noise. For further details see Campello et al. (2013).

The clusters discovered by HDBSCAN represent the regular ship behavior in the region of interest. The data clusters can, therefore, be used to create local models that describe the regular behavior for each cluster. The algorithm also functions as a form of pre-processing, where anomalous trajectories will be labeled as noise, as they do not correspond to any cluster of regular ship behavior. Predicting such anomalies is difficult, as the behavior of such vessels is often highly erratic. This study, therefore, focuses on modeling regular ship behavior, and discards the noise identified by HDBSCAN.

2.4. Classification Module

The classification module is trained using 30-minute segments of trajectory data. Each trajectory in the dataset is, therefore, split into 30 minute segments using a sliding window technique with a one minute interval. In this manner, the classification module will be trained using all possible 30 minute trajectory segments. Each segment is assigned a class label corresponding to the class of its parent trajectory, discovered via the clustering module. The objective of the classification module is to correctly classify an input trajectory segment to one of the underlying ship behavior clusters in the dataset.

The architecture of the classification module is illustrated in Fig. 6. Given that the trajectory segments consist of sequence data, it is suggested to utilize RNNs to encode the dynamic data. A bidirectional, stacked encoder is, therefore, utilized to encode the trajectory segments to a fixed size vector in a similar manner to the clustering module. The final hidden states of the backward and forward encoders are concatenated, allowing the encoder to preserve dependencies in both directions.

Historical AIS data, however, is not limited the dynamic data represented by the trajectory data. Static data is also available, e.g. ship type, ship length, date, etc. Assuming that this information is available via AIS at the time of prediction, such static data can be utilized by the classifier to discriminate between classes. In this study, it is suggested to include the ship type and length in the static data. Ship type will play a significant role in the behavior of a vessel, and should aid the classifier in achieving higher accuracy. The

ship length should also play a role in the type of behavior to be expected by the ship.

The static data is further separated into categorical data, s_{cat} , and continuous data, s_{cont} . The continuous features, e.g. length, can simply be scaled and concatenated with the dynamic data, as shown in Fig. 6. The categorical data, however, e.g. ship type, must be encoded using an embedding layer. An embedding layer maps a category to a vector representation. The concept was introduced to aid natural language processing, where word embeddings (Mikolov et al., 2013) provide the basis for many natural language processing tasks. In this study, it is suggested to embed the categorical static data, and concatenate these embeddings with the remainder of the data as shown in Fig. 6.

The concatenated data are then fed into fully connected, i.e. linear, layers. The final layer will have an output dimension corresponding to the number of classes, i.e. clusters, discovered by the clustering module. A softmax layer then computes the final output by scaling the output between 0 and 1 according to (14).

$$\hat{c}_i = \frac{e^{v_i}}{\sum_{j=1}^C e^{v_j}} \quad (14)$$

v_i is the predicted value for class i from the fully connected layers, and \hat{c}_i is the softmax output for class i . In this manner, a probability distribution is created over the number of classes. The classifier then compares the softmax output to the true class vector \mathbf{c} , where $c_i = 1$ for the true class and $c_j = 0 \forall j \neq i$. The cross entropy loss is then calculated and used to optimize the network.

When training the model, modern deep learning architectures, e.g. PyTorch (Paszke et al., 2019), include a softmax layer in the cross entropy loss. As a result, when training a network using the built in cross entropy loss, the softmax layer is not included in the architecture of the network. When evaluating the model, the predicted class is, therefore, generally taken as the argmax of \mathbf{v} without using a softmax layer. However, given that the softmax function gives a probability distribution over the number of classes, this can be used to identify a distribution over the ship behavior clusters the trajectory segment belongs to. As such, multiple trajectory clusters can be identified as possible for the selected vessel during a prediction. Therefore, a softmax layer is applied to the network during evaluation in this study.

2.4.1. Local Behavior Module

Given that the clustering module has discovered clusters of ship behavior in the historical AIS data, local models can be created to predict the ship behavior. Each cluster represents a group of localized ship behavior. Training a model on the subset of data corresponding to this local behavior should improve the predictive capabilities of the algorithm, as opposed to training on all available data.

In this study, each local model is comprised of a sequence-to-sequence model (Sutskever et al., 2014). The VRAE in

Sec. 2.3.1 is such an architecture, where a sequence-to-sequence model is utilized to aid in clustering. As outlined in Sec. 2.3.1, this encoder-decoder approach is common in natural language processing. The core of such models is a RNN, where the RNN used in this study is a GRU. The encoder RNN encodes the input sequence to a fixed size vector, and the decoder RNN decodes the target sequence using this vector, i.e. latent representation, of the input sequence. In an autoencoder architecture, as in Sec. 2.3.1, the target sequence is equal to the input sequence. In this manner the decoder's task is to reconstruct the input.

The local models in this study, however, take the past 30 minute behavior of a selected vessel as input, and predict the future 30 minute behavior. As such, the past 30 minutes must be encoded into a fixed size vector, and the future 30 minutes must be predicted using this representation. For an autoencoder, this bottleneck in the latent representation provides the basis for clustering, as one wishes to discriminate between classes in this space. When a sequence-to-sequence model is used for predictions, however, this bottleneck is detrimental to the performance.

The bottleneck in sequence-to-sequence models limits the capacity of the model, as an entire sequence must be predicted from a single vector. Furthermore, the encoder often becomes gradient starved. This is due to the fact that gradients calculated from the loss in the decoder must flow via the bottleneck during backpropagation. As a result, the encoder side of the network does not update well during training. Bahdanau et al. (2015) introduced an attention mechanism that addresses this issue. Instead of only looking at the final hidden state of the encoder, the decoder is able to look at all of the encoder hidden states, enhancing the predictive performance of the model. Using such an architecture, gradients are allowed to flow freely to the encoder side of the network via the attention mechanism. This approach was developed for translation tasks, but the architecture is also relevant for sequence-to-sequence tasks involving time series data.

The local model architecture in this study, therefore, utilizes the attention mechanism in Bahdanau et al. (2015) to facilitate effective ship trajectory prediction. The architecture of the local model is illustrated in Fig. 7. The attention mechanism is facilitated by a fully connected network, represented by the pink box in Fig. 7. The attention mechanism takes the previous hidden state of the decoder, i.e. \mathbf{h}_{t-1} , as well as all of the encoder hidden states as input. In this study, the encoder is bidirectional and stacked. As a result, the input to the attention mechanism will be the hidden states from the top layer, which contain the concatenated backward and forward hidden states of the encoder for each time step. The attention mechanism in this study functions in two steps. First, \mathbf{h}_{t-1} of the decoder is matched with the encoder hidden states. For the case of ship trajectory prediction, this can be thought of as how relevant ship behavior at some point during the past 30 minutes is for conducting a prediction at the current time step. The network can in this manner learn what to look at in order to most effectively conduct a prediction. The outputs are then run through a softmax layer

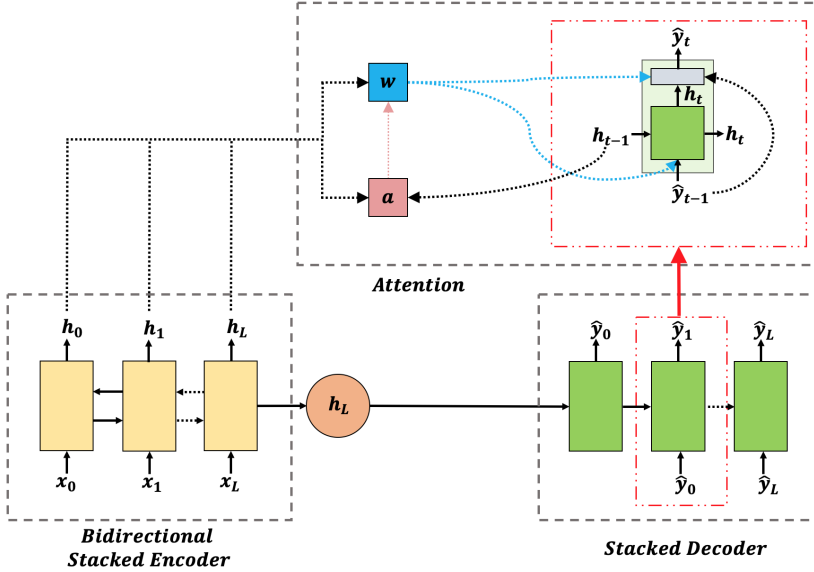


Figure 7: Prediction architecture using a sequence-to-sequence model with attention. The bidirectional stacked encoder is illustrated in yellow, and the stacked decoder in green. For each prediction in the decoder, the attention mechanism looks at the encoder hidden states and calculates a weighted sum. This is then input to the decoder RNN cell.

as in (14). This generates an attention distribution, $\hat{\mathbf{a}}$, over the encoder hidden states, where each attention value can be viewed as a weight for the corresponding encoder hidden state. A weighted sum of the encoder hidden states is then calculated, illustrated by the blue box in Fig. 7. The block takes in the encoder hidden states, as well as the attention weights, and outputs a weighted sum.

The architecture of each decoder cell is illustrated in the red box in the upper right of Fig. 7. The input to each RNN cell is a concatenation of the previous prediction, $\hat{\mathbf{y}}_{t-1}$, with the weighted encoder hidden states, \mathbf{w} . Furthermore, the linear layer that conducts the prediction for each state, $\hat{\mathbf{y}}_t$, takes $\hat{\mathbf{y}}_{t-1}$, and \mathbf{w} as input. Each prediction can, therefore, look at the entire past trajectory, and identify relevant parts to conduct as accurate a prediction as possible. The linear layer is allowed to look at the current input, to further enhance the accuracy of the prediction, where short term dependencies can be directly determined. For the case of ship trajectory prediction, the next state will undoubtedly have a high dependency on the previous.

Each local model is, therefore, trained using the outlined architecture. The decoder will function in the same manner as in Sec. 2.3.1, where states are iteratively predicted, but instead of reconstructing the input, a target sequence, \mathbf{y} is predicted. The loss is calculated using the mean squared error as in Sec. 2.3.1. To optimally train the network, all combinations of past and future 30 minute trajectory segments should be utilized. In this study, one hour trajectory segments were extracted using a sliding window technique, where the window size was one minute. The first 30 minutes

of each trajectory are defined as the input (i.e. past), and the final 30 the target (i.e. future).

3. Results and Discussion

In this section, the results from a case study using the outlined deep learning framework are presented. A dataset corresponding to one year of AIS data from January 1st 2017 to January 1st 2018 for the region around the city of Tromsø, Norway was utilized. This region contains complex traffic, and provided a relevant test case for the framework.

PyTorch (Paszke et al., 2019) was utilized to implement the neural networks. All networks were trained using the Adam optimizer (Kingma and Ba, 2015). Furthermore, gradient clipping (Pascanu et al., 2013) and batch normalization (Ioffe and Szegedy, 2015) were utilized to aid in convergence. Hyperparameters were tuned for this specific region, and will need to be tuned to the specific geographical region to which the framework is to be applied.

3.1. Clustering Module

In this section, the results for the clustering module are presented. The technique described in Sec. 2.3 was applied to the dataset corresponding to the region surrounding Tromsø. This corresponded to approximately 70,000 trajectories. In this study, a VRAE was utilized to cluster the historical AIS trajectories. The results indicated that using a VRAE as opposed to a β -VRAE resulted in more optimal clusters for the architecture and data in this study, i.e. $\beta = 1$. It appeared based on visual inspection of the clusters of trajectories, that increasing the value of β caused multiple local

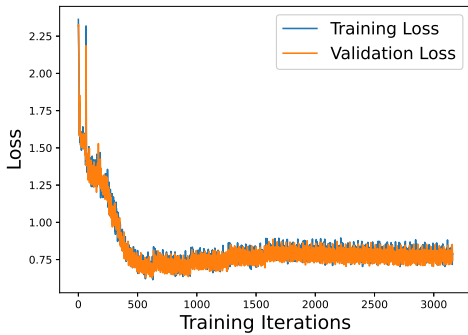


Figure 8: VRAE loss.

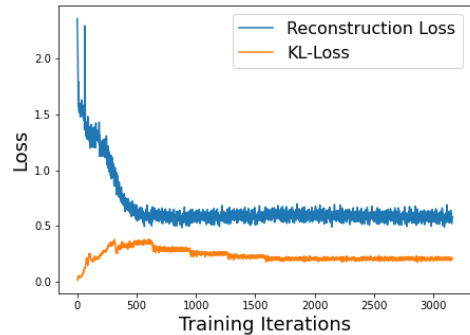


Figure 9: Reconstruction and KL losses for VRAE.

behavior clusters to merge. This may degrade the results of the subsequent trajectory prediction, as it is desirable to discover behavior clusters that are as specific as possible. It should be noted that the optimal value of β will vary based on the complexity of the ship traffic in the region of interest.

3.1.1. Network Training

The VRAE utilized in this study was comprised of a stacked, bidirectional GRU encoder with 3 layers, each with a hidden size of 50. A stacked GRU decoder with 2 layers, and a hidden size of 50 was used. The dimensionality of the latent space was set to 20 to allow for further compression of the data. A number of variations of parameters were run to determine the best performance.

During training, the dataset was shuffled and split into training and validation datasets. The training data accounted for 90% of the trajectories, and the validation 10%. Fig. 8 illustrates the total loss of the VRAE on both the training and validation sets during training for 10 epochs. The results indicate that the model is not overfitting to the data, as the validation and training losses are highly correlated for the duration of the training. It appears that the total loss increases over time, but this effect is due to a technique known as KL-annealing, where the KL-loss term is introduced linearly over a span of a number of epochs. In this study, it was introduced over five epochs, as can be seen in Fig. 9. In this figure, the reconstruction- and KL-loss terms are plotted individually. KL-annealing allows the model to learn how to reconstruct the data before enforcing the KL-regularization term. As a result, it can be seen that the reconstruction loss decreased quickly, whilst the KL-term increased. Each step of the KL-loss downwards after this corresponded to an increase in its weighting. The loss terms converged after this, and it was concluded that the model had converged.

3.1.2. Clustering Results

In order to cluster the trajectories, a forward pass of the encoder was run to generate the latent representations for each trajectory. The trajectories are then clustered in this space using HDBSCAN. The implementation in McInnes

et al. (2017) was utilized in this study.

The minimum cluster size in the algorithm, however, is found to be decisive in the type of clusters discovered. This value was varied, and found to play a significant role in the outcome of the remainder of the architecture. When the minimum cluster size was set to 10, over 400 clusters of vessel behavior were discovered. In this case, the algorithm is able to discover very specific vessel behavior. This is beneficial as one wishes to discriminate between behavior clusters, and generate predictions using these clusters. More specific behavior should lead to better predictions. However, when classifying a 30 minute trajectory segment to one of these clusters in the classification module, the network will have great difficulty in correctly classifying the segment, and the performance of the overall algorithm will be degraded. This is due to the existence of many similar clusters, such that the algorithm is unable to conduct an accurate classification.

Increasing the minimum cluster size causes clusters of local behavior to merge into larger clusters, where the behavior within a given cluster varies to a greater degree. Discovering smaller clusters allows the model to discover a greater number of ship speed clusters within a given route for instance. Merging these clusters, however, allows the model to classify a given trajectory segment with a higher degree of accuracy, contributing to the overall success of the algorithm. As a result, a minimum cluster size of 100 was set based on trials for various minimum cluster sizes. In this case, 52 ship behavior clusters were discovered. Fig. 10 illustrates the clustered latent representations, where it appears that the algorithm had discovered meaningful clusters. Fig. 11 illustrates a subset of the clusters discovered by the module. Each of these clusters corresponds to a cluster of historical ship behavior. Despite utilizing clusters of more general behavior, discovering such main local ship behavior cluster will aid the performance of the local prediction module, which, with its architecture, can predict various behavior within these main clusters.

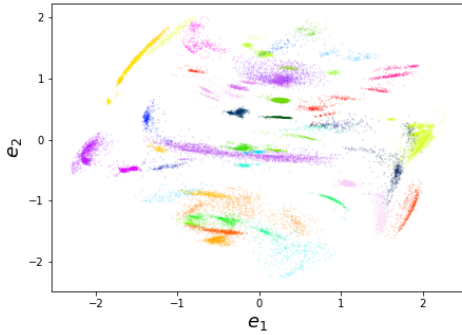


Figure 10: Clusters in latent space of VRAE. The latent space is illustrated using the top two principle components of the space, e_1 and e_2 . Clusters with similar colors are not necessarily the same.

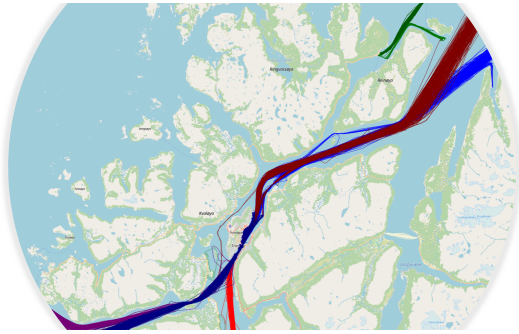


Figure 11: Subset of discovered trajectory clusters with minimum cluster size of 100.

3.2. Classification Module

The classification module was designed with a bidirectional, 5-layer stacked GRU with 20 hidden units as the encoder. 3 linear layers were utilized as the classification head, where the first was set to have one fourth as many neurons as the number of classes (i.e. number of clusters), the second half as many, and the third as many neurons as the number of classes. The embedding size was set to 10. During training, it was found that embedding the ship type led to the model focusing too much on the ship type, thereby degrading the results. As a result, a dropout rate of 50 % (Srivastava et al., 2014) was applied to the embedding layer to prevent overfitting to the embedding data.

3.2.1. Network Training

Prior to training the network, the dataset was shuffled and split into training (70 %), validation (10%) and test (20%) sets. Subsequently, each dataset was split into 30 minute trajectory segments using a sliding window technique with a window size of one minute. As a result, all possible 30 minute trajectory segments in the data are used to train the

Table 1

Size of datasets.

Training	Validation	Test
1.127×10^6	1.66×10^5	3.16×10^5

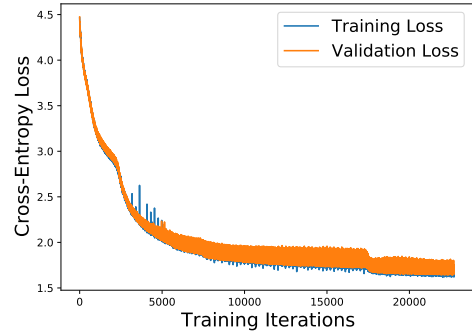


Figure 12: Classification network loss, where the loss is defined as the cross-entropy loss.

classifier. The size of the datasets is shown in Table 1.

The results of the training are illustrated in Fig. 12. It appears that both the training and validation losses continue to decrease until about 20000 iterations. As a result, it was concluded that the model had converged at this point. Furthermore, the validation loss is closely correlated with the training loss. Therefore, it was concluded that the model is not overfitting to the data.

3.2.2. Classification Results

When evaluating the classification performance on the test set, the classification accuracy was found to be 47 %. However, in this study it is suggested to use the softmax distribution of possible clusters. Any clusters with a softmax output over 0.1 (i.e. 10% probability) are output as likely ship behavior clusters. In this manner, the model identifies multiple possible clusters the trajectory segment may belong to. The softmax distribution for a randomly selected vessel trajectory segment from the test set is illustrated in Figure 13. The colors of the bars correspond to the colors of the trajectory clusters illustrated in Fig. 14.

For the selected vessel in Figure 13, the model correctly classified the behavior to cluster 25 (i.e. blue). However, the softmax output also indicated that the selected vessel may belonged to cluster 23 (i.e. red), as its probability was above 0.1. Furthermore, cluster 24 had a probability above 0.05, and might have been a possible behavior cluster. All three clusters share common behavior, and in this case, it appeared appropriate to investigate multiple possible clusters.

When using a 10 % softmax threshold, the classification accuracy increased to 73 % for the test set. On average, the model outputs 3 possible clusters a trajectory segment may belong to. Decreasing the threshold was also investigated.

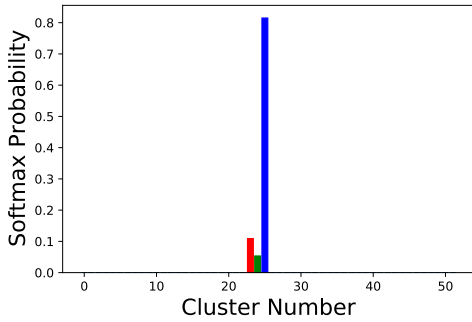


Figure 13: Softmax probability distribution for selected vessel case, illustrating uncertainty of cluster assignment.

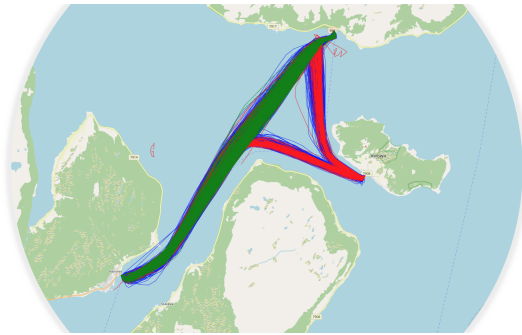


Figure 14: Classified ship behavior clusters for selected vessel case. The selected vessel was classified to the blue, with the red above the softmax threshold of 10%, and the green above 5%.

Table 2
Classification Accuracy and Average Number of Clusters.

	Absolute	10% Softmax	5% Softmax
Accuracy	47%	73%	92%
Clusters	-	3	5

With a threshold of 5 %, the accuracy rate increased to 92 %. In this case, the model outputs an average of 5 possible clusters the trajectory segment may belong to. The results are summarized in Table 2. For the selected vessel case in Fig. 13 and 14, the blue and red clusters would be identified for a softmax threshold of 10%, with the addition of the green cluster for a threshold of 5 %.

Overall, it appears that the model was successful in classifying the trajectory segments in the test set, where the accuracy increased as the softmax probability threshold was lowered.

3.3. Local Behavior Module

In the local behavior module, local models for each cluster of ship behavior are available. In this study, however,

Table 3
Size of Local Behavior Model Datasets.

	Training	Validation	Test
Model 1	9.23×10^4	2.80×10^4	1.2×10^4
Model 2	1.02×10^4	3.3×10^3	1.3×10^3

it was infeasible to train 52 neural networks to evaluate the overall performance using the available resources. In a commercial setting, however, this should be done. As such, only two models were trained to illustrate the performance of the method. These correspond to the blue and red clusters from the example in Sec. 3.2.2, illustrated in Fig. 14. These models are chosen as they were above the 10% softmax threshold used in this study. These models are hereafter referred to as model 1 (i.e. the blue cluster), and model 2 (i.e. the red cluster).

Both models have a bidirectional 2-layer stacked GRU encoder with 20 hidden units, and a 2-layer stacked GRU decoder with 20 hidden units. Variations of these architectures were evaluated to determine the architecture with the best performance.

3.3.1. Network Training

The datasets for both models were initially reduced to only contain the trajectories in the respective clusters. Subsequently, each model dataset was shuffled and split into training (70 %), validation (10%) and test (20%) sets. These datasets were again split into 30 minute trajectory segments using a sliding window technique with a window size of one minute. Source sequences (past 30 minute trajectory), and their corresponding target sequences (future 30 minute trajectory) were extracted in this manner. The sizes of the respective datasets are summarized in Table 3.

The training and validation losses for model 1 and model 2 are illustrated in Fig 15 and Fig. 16, respectively. Both models were trained for 1000 epochs. The training and validation losses were both correlated for the duration the training of the models, indicating that neither model overfit to the data. The losses continue to decrease for the duration of the training iterations illustrated in the figures. Once the decrease was minimal, the models were assumed to have converged.

3.3.2. Prediction Results

The predictive performance of the models was evaluated on their respective test sets. The results for each model are presented in Figure 17. The figure illustrates the root mean squared error (RMSE) of the predicted position as a function of the prediction horizon. The results indicate that model 2 has better performance, with a mean squared error of 436 m for a prediction horizon of 30 minutes, whilst model 1 has a mean squared error of 576 m. This may be due to model 1 being trained on a greater number of trajectories. The cluster for model 1 was much larger than for model 2. As a result, there will be a greater degree of variation in the data, and the model is not as effective in capturing the variation. In model

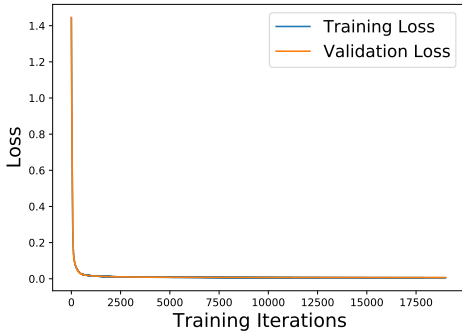


Figure 15: Model 1 training and validation loss.

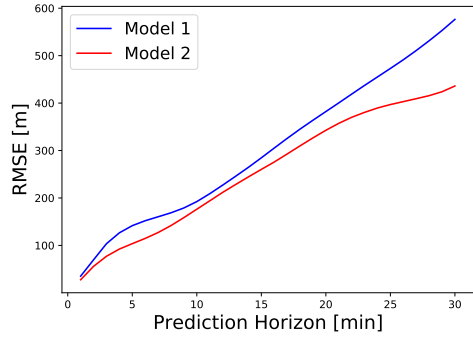


Figure 17: Root mean squared error (RMSE) of test set predictions.

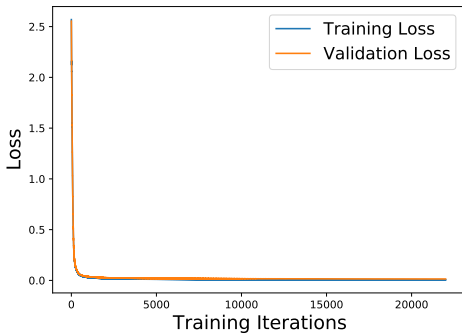


Figure 16: Model 2 training and validation loss.

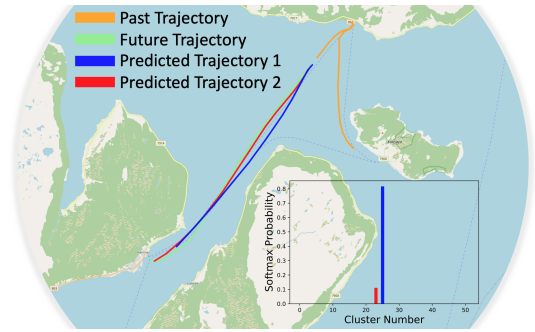


Figure 18: Predictions for selected vessel case.

2, the data likely has less variation, and, therefore, fits well to the data.

Figure 18 illustrates the prediction results for the selected vessel case in Sec. 3.2.2. Here, two predictions were conducted using the two models identified by the classification module. The results indicate the the sequence-to-sequence model with attention can provide successful results, even with highly nonlinear input trajectories. It appears that the attention mechanism allows the models to focus on the most relevant aspects of the past trajectory. Model 2, however, appears to have better performance than model 1. Model 1 should have better performance, as the test trajectory belongs to the cluster for model 1. Nonetheless, in this case it appears that the model with the second highest probability has the best performance. This indicates that incorrectly classifying the selected vessel may not result in a significant degradation in the prediction results. As such, many of the incorrectly classified trajectory segments may be classified to a cluster of similar behavior.

4. Conclusion and Further Work

Limited work has been conducted on utilizing deep learning to enhance the safety of maritime transportation systems. One area of interest is in aiding maritime situation awareness via proactive collision avoidance. To facilitate this, global scale trajectory predictions, ie. order 5-30 minutes, must be conducted. This study suggests an approach to this issue via a deep learning framework for regional trajectory prediction. Using such an architecture, the future trajectory of a vessel can be predicted in under a second. The framework investigates utilizing modern deep learning architectures to facilitate a decomposition of regional ship behavior into local models. Utilizing historical AIS data, the framework is successful in clustering historical ship behavior using a variational recurrent autoencoder. The results also indicate that the increased complexity of the model allows it to cluster vessel behavior more successfully.

Utilizing this historical knowledge, the past behavior of a selected vessel is classified to the most likely clusters of historical behavior. Predictions corresponding to the behavior in each cluster are then output to the user. The local prediction models are comprised of sequence-to-sequence models with attention. The results indicate that the attention mech-

anism assists the prediction by allowing the model to focus on the most relevant parts of the past trajectory. By decomposing the behavior into local models, greater accuracy can be achieved than training a similar prediction model to the data in all clusters. Overall, the suggested framework is successful in predicting trajectories on a global scale.

Further work will include providing further uncertainty estimation via Bayesian dropout techniques. In this manner, a distribution is predicted for each time step. The classification module will also be further improved to enhance the classification accuracy. Weather parameters will likely aid the predictions, as ships will display various behavior based on the prevailing weather conditions. This should also be addressed in future work.

Acknowledgements

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, Industry and Fisheries. The authors would also like to express gratitude to the Norwegian Coastal Administration for providing access to their AIS database. Map data copyrighted OpenStreetMap contributors and available from <https://www.openstreetmap.org>.

References

- Bahdanau, D., Cho, K.H., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. arXiv:1409.0473.
- Bengio, Y., Simard, P., Frasconi, P., 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 5, 157–166. doi:10.1109/72.279181.
- Campello, R.J.G.B., Moulavi, D., Sander, J., 2013. Density-based clustering based on hierarchical density estimates, in: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (Eds.), *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 160–172.
- Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches URL: <http://arxiv.org/abs/1409.1259>, arXiv:1409.1259.
- Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, in: NIPS'2014 Deep Learning workshop. URL: <http://arxiv.org/abs/1412.3555>, arXiv:1412.3555.
- 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.
- Daranda, A., 2016. Neural Network Approach to Predict Marine Traffic. Technical Report 3.
- Du, L., Goerlandt, F., Kujala, P., 2020. Review and analysis of methods for assessing maritime waterway risk based on non-accident critical events detected from AIS data. doi:10.1016/j.res.2020.106933.
- Endsley, M.R., Bolté, B., Jones, D.G., 2003. *Designing for situation awareness: an approach to user-centered design*. Taylor & Francis.
- Fabius, O., van Amersfoort, J.R., 2015. Variational Recurrent Auto-Encoders, in: Proceedings of the International Conference on Learning Representations (ICLR). URL: <http://arxiv.org/abs/1412.6581>.
- Forti, N., Millefiori, L.M., Braca, P., Willett, P., 2020. Prediction of Vessel Trajectories from AIS Data Via Sequence-To-Sequence Recurrent Neural Networks, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Institute of Electrical and Electronics Engineers Inc., pp. 8936–8940. doi:10.1109/ICASSP40776.2020.9054421.
- Goerlandt, F., Kujala, P., 2011. Traffic simulation based ship collision probability modeling, in: *Reliability Engineering and System Safety*, Elsevier. pp. 91–107. doi:10.1016/j.res.2010.09.003.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press.
- Graves, A., Mohamed, A.R., Hinton, G., 2013. Speech recognition with deep recurrent neural networks, in: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, pp. 6645–6649. doi:10.1109/ICASSP.2013.6638947, arXiv:1303.5778.
- Hexeberg, S., Flåten, 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. doi:10.23919/ICIF.2017.8009762.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A., 2017. beta-vae: Learning basic visual concepts with a constrained variational framework, in: Proceedings of the International Conference on Learning Representations (ICLR).
- Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation* 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735.
- Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: 32nd International Conference on Machine Learning, ICML 2015, International Machine Learning Society (IMLS). pp. 448–456. URL: <https://arxiv.org/abs/1502.03167v3>, arXiv:1502.03167.
- Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations (ICLR). arXiv:1412.6980.
- Kingma, D.P., Welling, M., 2014. Auto-Encoding Variational Bayes, in: Proceedings of the International Conference on Learning Representations (ICLR). URL: <http://arxiv.org/abs/1312.6114>, arXiv: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. doi:10.1109/SDF.2015.7347707.
- McInnes, L., Healy, J., Astels, S., 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software* 2, 205.
- Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space, in: 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, International Conference on Learning Representations, ICLR. arXiv:1301.3781.
- Murray, B., Perera, L.P., 2020a. A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data. *Ocean Engineering* 209, 107478. doi:10.1016/j.oceaneng.2020.107478.
- Murray, B., Perera, L.P., 2020b. Deep Representation-Based Vessel Trajectory Clustering for Situation Awareness in Ship Navigation, in: Proceedings of the 5th International Conference on Maritime Technology and Engineering (MARTECH 2020).
- Nguyen, D., Vadaine, R., Hajdich, G., Garello, R., Fablet, R., 2018. A multi-task deep learning architecture for maritime surveillance using ais data streams, in: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), pp. 331–340.
- 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. *Information Fusion (FUSION)*, 2014 17th International Conference on , 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, 2218–2245. doi:10.3390/e15062218.
- Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. 30th International Conference on Machine Learning, ICML 2013 , 2347–2355 URL: <http://arxiv.org/abs/1211.5063>, arXiv:1211.5063.

- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library, in: Wallach, H., Larochele, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035. URL: <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library> pdf.
- Perera, L.P., 2017. Navigation vector based ship maneuvering prediction. *Ocean Engineering* 138, 151–160. doi: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. *Advanced Ship Design for Pollution Prevention*, 205–216doi:10.1201/b10565-26.
- Perera, L.P., Guedes Soares, C., 2015. Collision risk detection and quantification in ship navigation with integrated bridge systems. *Ocean Engineering* 109, 344–354. doi:10.1016/j.oceaneng.2015.08.016.
- Perera, L.P., Murray, B., 2019. Situation Awareness of Autonomous Ship Navigation in a Mixed Environment Under Advanced Ship Predictor, in: *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, American Society of Mechanical Engineers (ASME). doi: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 Transactions on Intelligent Transportation Systems* 13, 1188–1200. doi:10.1109/TITS.2012.2187282.
- Rezende, D.J., Mohamed, S., Wierstra, D., 2014. Stochastic backpropagation and approximate inference in deep generative models, in: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*.
- Ristic, B., Scala, B.L., Morelande, M., Gordon, N., 2008. Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction. *2008 11th International Conference on Information Fusion*, 40–46doi:10.1109/ICIF.2008.4632190.
- Rong, H., Teixeira, A., Guedes Soares, C., 2019. Ship trajectory uncertainty prediction based on a Gaussian Process model. *Ocean Engineering* 182, 499–511. doi:10.1016/j.oceaneng.2019.04.024.
- Rong, H., Teixeira, A.P., Guedes Soares, C., 2020. Data mining approach to shipping route characterization and anomaly detection based on AIS data. *Ocean Engineering* 198. doi:10.1016/j.oceaneng.2020.106936.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature* 323, 533–536. doi:10.1038/323533a0.
- Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 2673–2681. doi:10.1109/78.650093.
- Silveira, P., Teixeira, A., Soares, C.G., 2013. Use of AIS Data to Characterise Marine Traffic Patterns and Ship Collision Risk off the Coast of Portugal. *Journal of Navigation* 66, 879–898. URL: http://www.journals.cambridge.org/abstract/_jS0373463313000519, doi:10.1017/S0373463313000519.
- Srivastava, N., Hinton, G., Krizhevsky, A., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Technical Report. doi:10.5555/2627435.2670313.
- Srivastava, N., Mansimov, E., Salakhutdinov, R., 2015. Unsupervised Learning of Video Representations using LSTMs. *32nd International Conference on Machine Learning, ICML 2015* 1, 843–852. URL: <http://arxiv.org/abs/1502.04681>, arXiv:1502.04681.
- Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems* 4, 3104–3112. URL: <http://arxiv.org/abs/1409.3215>, arXiv:1409.3215.
- 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 Transactions on Intelligent Transportation Systems*, 1–24doi:10.1109/TITS.2017.2724551.
- Xiao, Z., Fu, X., Zhang, L., Goh, R.S.M., 2020. Traffic Pattern Mining and Forecasting Technologies in Maritime Traffic Service Networks: A Comprehensive Survey. doi:10.1109/TITS.2019.2908191.
- Yao, D., Zhang, C., Zhu, Z., Huang, J., Bi, J., 2017. Trajectory clustering via deep representation learning, in: *Proceedings of the International Joint Conference on Neural Networks, Institute of Electrical and Electronics Engineers Inc.* pp. 3880–3887. doi:10.1109/IJCNN.2017.7966345.
- Zhang, W., Feng, X., Goerlandt, F., Liu, Q., 2020. Towards a Convolutional Neural Network model for classifying regional ship collision risk levels for waterway risk analysis. *Reliability Engineering and System Safety* 204, 107127. doi:10.1016/j.res.2020.107127.