VILNIUS UNIVERSITY

Julius
VENSKUS

Semi-supervised and Unsupervised Machine Learning Methods for Sea Traffic Anomaly Detection

DOCTORAL DISSERTATION

Technological Sciences
Informatics Engineering T 007

Vilnius 2021

This dissertation was written between 2016 and 2020 in Vilnius University.

Academic supervisor:
Assoc. Prof. Dr. Povilas Treigys (Vilnius University, Technological Sciences, Informatics Engineering T 007).

Academic consultant:
Prof. Dr. Arūnas Andziulis (Klaipeda University, Technological Sciences, Informatics Engineering T 007).

VILNIAUS UNIVERSITETAS

Julius
VENSKUS

Dalinai prižiūrimų ir neprižiūrimų mašininio mokymosi metodų tyrimas jūrų eismo anomalijoms aptikti

DAKTARO DISERTACIJA

Technologijos mokslai
Informatikos inžinerija T 007

Vilnius 2021

Disertacija rengta 2016–2020 metais Vilniaus universitete.

Mokslinis vadovas:
doc. dr. Povilas Treigys (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija - T 007).

Mokslinis konsultantas:
prof. dr. Arūnas Andziulis (Klaipėdos universitetas, technologijos mokslai, informatikos inžinerija - T 007).

Acknowledgements

First and foremost, praises and thanks to the God for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my sincere gratitude to my advisor Doc. Dr. Povilas Treigys for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

My sincere thanks also goes to Dr. Jolita Bernataičienė and Dr. Jurgita Markevečiūtė for giving very valuable advice.

Acknowledgment. The author is thankful for the high-performance computing resources provided by the Information Technology Open Access Center at the Faculty of Mathematics and Informatics of Vilnius University Information Technology Research Center.

Contents

List of Figures

List of Tables

Acronyms

AIS Automatic Identification System. 3, 4, 18, 21–23, 60, 62, 90

API Application Programming Interface. 18

COG Course Over ground. 25, 26, 34, 37

ECMWF European Centre for Medium-Range Weather Forecasts. 18

EDA Exploratory Data Analysis. 5

GMM Gaussian Mixture Model. viii, 7, 10, 43, 65

IMO International Maritime Organization. 19

LSTM Long Short Term Memory. x, 5–7, 17, 24, 37, 39, 41, 65, 66, 68, 84, 91–94

MDS Multi Dimensional Scaling. 5, 26, 34

MLP Multi Layer Perceptron. x, 5, 38, 39

MMSI Maritime Mobile Service Identity. 19, 21, 26, 27

MSA Maritime Situational Awareness. 2, 3, 5, 6, 20, 29

NATO North Atlantic Treaty Organization. 2

PICP Prediction Region Coverage Probability. 5, 67–69, 73, 74, 91–93

PINAW Prediction Region Normalized Average Width. 5, 67–69, 91

ReLU Rectifier Linear Unit. 38

ROT Rate of Turn. v, 23, 25

SMOTE Synthetic Minority Oversampling Technique. 5, 36, 38, 61

SOG Speed Over ground. v, 23, 25, 34, 37

SOM Self-Organizing Map. v, viii, 5–7, 10, 24, 43–46, 49, 51, 65, 78, 80, 82–86, 90, 91, 93, 94

SOM_GMM Self-Organizing Map (SOM) with integrated Gaussian Mixture Model (GMM). 7, 11, 84, 85, 93

SOM_pheromone Self-Organizing Map (SOM) neural network with integrated virtual pheromone. 7, 11, 84, 93

TANH Hyperbolic Tangent Function. 65

UTM Universal Transverse Mercator. 24

VHF Very High Frequency. 23

VTS Vessel Traffic Service. 3, 6, 7, 18, 20, 22, 29, 84

WGS84 World Geodetic System 1984. 24, 33, 36

List of Symbols

The next list describes several symbols that will be later used within the body of the document

Y a matrix of vessel navigational sequences used as the model output data, the true values.

χ a matrix of vessel navigational sequences used as the model input data.

$\hat{Y}$ a predicted / reconstructed output sequences of vessel navigational vector sequence by model.

β_{PPV} an influence weight for classification precision of SOM virtual pheromone

β_{TPR} an influence weight for classification sensitivity of SOM virtual pheromone

Δt a time interval in seconds of successive navigation vectors.

η a sliding window predefined step used in producing of vessel navigational vectors sequences.

η_{ij} a neighbourhood radius of SOM neighbourhood function

$\hat{S}$ a sample covariance matrix.

λ a tuneable parameter that represents the relative importance of proposed classical and region loss functions.

ρ a virtual pheromone intensity evaporation speed

τ a current sliding window position in single vessel sequence.

τ_{ij} a virtual pheromone intensity in SOM neuron

Θ a threshold value of SOM virtual pheromone classification error rate

$\widetilde{n}$ a sequence length for neural network output.

$c(t)$ a LSTM cells transmitted cell state at time step t.

d_{ij} a distance between the current observation vector and the winning neuron

$e^{(l)}_{(g,i,j)}$ a error of a reconstruction of single navigational vector's feature, where j - jth navigational vector's feature, f - number of features, l - model type.

f a number of fields/features in vessel navigational vector before data preprocessing.

f' a number of fields/features in vessel navigational vector after data preprocessing.

$F()$ a step function of SOM neighbourhood function

$F_{d_1,d_2}(a)$ an a-level critical value of a Fisher distribution with d_1 and d_2 degrees of freedom.

FN false negatives

FP false Positives

$h(t)$ a hidden state of LSTM cell at time step t.

$J(\Theta)$ a classification error rate

k a number of wild bootstrapping replicates of network trainings.

l a type of the model (upper, crisp, or lower).

L^l_s a common part of the loss function for l type of model.

$L^{(crisp)}_{total}$ the overall loss function for crisp model.

$L^{(lower)}_{\ell}$ a specific loss function for lower bound.

$L^{(lower)}_{total}$ an overall lower loss function.

$L_{\ell}^{(upper)}$ a specific loss function for upper bound.

$L_{total}^{(upper)}$ an overall upper loss function.

N a number of training sequences in the training data set.

N' a number of vessels navigational records, data vectors in raw data set.

n' a sequence length for input to neural network.

p a dimension number of prediction vector.

P_v a number of navigational vectors of single particular vessel.

p_v a individual vessel navigational vector index in data set, where $p_v \in \{1, 2, \ldots, P_v\}$

$PICP$ a prediction region coverage probability.

$PINAW$ a prediction region normalized average width.

PPV_{Θ} a classification precision of SOM virtual pheromone

Q_{lstm} a number of Long Short Term Memory (LSTM) cells in a layer

Q_{mlp} a number of Multi Layer Perceptron (MLP) neurons in a hidden layer

r a step ahead of a LSTM network predictions, where $r \in \{1, 2, , \ldots, m\}$

$ReLU()$ a rectified linear unit function.

S a restructured navigational data vectors of each ship into matrix rows according to the ship's field/feature $x^{(MMSI)}$

$T_{\Delta t}$ a time interval threshold for dropping/filtering out vessel's sequence in data cleaning process.

T_{acc} a critical threshold value of predictive model in order to fill missing value.

TN true Negatives

TP true Positives

TPR_{Θ} a classification sensitivity of SOM virtual pheromone

V a number of vessels in data set

v a vessel index in data set, where $v \in \{1, 2, \ldots, V\}$

W a two-dimensional array of SOM neurons

X a raw data set of vessel navigational vectors/records.

$x(t)$ a LSTM cell input at time step t.

$x^{(j)}$ where $g \in \{1, 2, \ldots, f\}$ vessel navigation vector features as: vessel unique identifier MMSI, 'Latitude', 'Longitude', SOG, COG, Ship type, Timestamp of the data being received.

x_g where $g \in \{1, 2, \ldots, N'\}$ is a single vessel navigational data record/vector that consist of $x^{(j)}$ parameters/features, where $g \in \{1, 2, \ldots, f\}$

$y(t)$ a LSTM cell output at time step t.

Introduction

The maritime logistics industry is a crucial component of the global trade economy with expanding volume, traffic intensity, and requirements. In Q1-Q3, 2019, 2,660 million tons gross weight of seaborne goods were handled in EU-27 main ports [1]. That is 7% more in comparison with the same quarters in 2016. Totally, more than 90% of cargo is transported by sea [2] in Europe. The industry is a critical and hazardous area of human activity and its growth raises control and security challenges. Increasing intensity in maritime traffic creates an increasing requirement for better prevention-oriented incident management systems. One of the control techniques of this complex management system is the detection of abnormal vessel movement. Detection is based on predicting vessel trajectories by analysing navigational data sequences and searching for irregular, illegal, and other anomalous appearances in trajectory/navigational data [3]. A maritime trajectory/navigational data in the form of a sequence of navigational vectors can include vessel geographical position, traffic parameters (e.g., speed and rotation), vessel entity identification numbers, and auxiliary data (e.g., meteorological data). Such a data set presents a large scale, complex data structure that has all necessary information for automated vessel traffic prediction and automated prediction evaluation to decide whether the traffic is normal or abnormal in the monitored sea area. For marine traffic monitoring, automated data gathering systems (e.g., Automatic Identification System, AIS) provide huge trajectory/navigational data sets for vessels. The sets are challenging for human-based analysis and anomaly detection [4]. In regard to the vessel movement prediction, the task becomes unsolvable without the application of algorithms. To solve the issue, agent-based, hybrid modeling, machine learning-based data analysis and data mining techniques is a promising techniques for this type of task. Observed patterns in data could help to forecast vessel movement based on previous trajectory data of vessels and make movement predictions under specific traffic and weather conditions. However, the fact that the vessels behave differently in different geographical sea regions, sea ports and their behaviour depends on the vessel type as well, which aggravates the task. These assumptions have to be incorporated in the investigation of anomaly detection approach.

Marine traffic is a dynamic system, where the traffic properties of a vessel change in space and time. The traffic data sets determine a structure that represents trajectories of multiple vessels. A trajectory of a single vessel consists of its position in space and other properties such as heading, course over the ground, etc. Typically, marine traffic data is collected and structured by AIS and can be viewed as a set of particular vessels data vectors representing vessel properties, geographical location, etc. at certain time moments, i. e. they can be viewed as spatio-temporal time series. Spatio-temporal data analysis is a challenging task for classical machine learning methods because behavioural patterns should capture vessel's position in relation to both space and time. Recently published works take advantage of extended LSTM (Long Short Term Memory) neural networks to learn spatio-temporal dependencies (see [5, 6, 7]) and offer promising techniques for further investigation.

Statement of the Problem

Maritime Situational Awareness (MSA) concept was presented by North Atlantic Treaty Organization (NATO) in their summit in Riga in 2006 as an extension of Maritime Domain Awareness (MDA). Couple years later NATO presented MSA Concept Development Plan. Its main purpose is to implement a MSA with a Doctrine, Organization, Training, Logistic, Leadership, Personnel, Infrastructures and Interoperability (DOTMLPII) approach [8].

The main goal of Maritime Situational Awareness (MSA) is to obtain a complete picture in Marine Domain by receiving information from multiple monitoring, surveillance, and reconnaissance systems, including knowledge extraction subsystems. Martineau and Roy state that "all aspects of a situation of interest in a timely manner, one can then say that complete and continuous situational awareness has been achieved" [9]. On the other hand, the final state of such goal is unreachable due to complexity and variability of the maritime domain. That understanding is supported by the same authors Martineau and Roy by stating it "would be akin to omniscience and achieving it would be a utopia" [9]. Continuous and timely data from multiple sources must be collected to obtain a clear picture of a situation. Additionally, pattern identification and extraction from the same data must be performed. Knowledge extraction is an essential system part of enriching MSA.

Safety and security have an essential role in marine domain. The MSA enables marine and coastal authorities to evaluate potential security and safety risks and take timely actions to mitigate these risks [8]. The high intensity of marine traffic and data generated by it makes it impossible for human cognitive abilities to be aware of situation. The data collection automation and knowledge extraction methods and their practical application in MSA might help authorities to pursue those goals [10]. Extraction of marine vessel behavioural patterns and evaluation hazardous situation of safety or security infraction are among the most important goals in MSA. Collection of large quantities of diverse data and knowledge extraction help coastal authorities to make well-founded decisions. [11, 12, 13]. One of the ways to enhance MSA is the identification of anomalous behaviour in marine traffic data (anomaly detection) [14, 15], that is strongly supported by multiple civilian, military, and law enforcement authorities around the world [14].

Definition of Anomaly

"Anomaly", "abnormal" and "anomaly detection" concepts can be found in various research fields such as fault diagnosis, video surveillance, network security, human activity monitoring, maintenance, etc. [14]. Ekman and Holst argue: "anomaly detection says nothing about the detection approach and it actually says nothing about what to detect" [16]. In multiple papers "anomaly detection" is presented from a human or computational perspective. There is both richness and vagueness in its meanings [14, 17].

In most of data-driven research, an anomaly is defined as a representation of deviation from normality. It follows Portnoy *et al.* definition: "anomaly detection approaches build models of normal data and then attempt to detect deviations from the normal model in observed data" [18]. While analyzing marine traffic, multiple papers describe the abnormal vessel movement slightly differently but mostly define it as an unreasoned movement deviation from the sea lanes, navigational routes, trajectory, speed, or other traffic parameters [11, 19, 4].

In this dissertation, anomaly detection is studied as a means of enhancement of Maritime Situational Awareness (MSA) as an active research area. In computer science and sea security, the term "anomaly" is used interchangeably with the same meaning and definition as "abnormal", "abnormal traffic", "anomalous traffic", etc.

The dissertation effort is devoted to research and development in the area of anomaly detection. The primary purpose is the security of the population. However, algorithms and methods of anomaly detection already exist, and the research in this dissertation focuses mostly on improving existing technologies. The anomaly detection gives a capability that enables authorities to prevent harmful events, and in case that is impossible, to prepare for them. These detections must be made as early as possible to give time for VTS to coordinate appropriate actions. The anomaly detection algorithms must be enriched for the detection of new types of anomalies.

Research Object

Detection of marine traffic anomalies in AIS data.

Research Aim And Objectives

The aim of the research is to investigate existing approaches and solutions and to propose a complex systemic (or integrated) approach including improvement of ML algorithms for detection of marine vessel traffic anomaly in AIS data.

For this aim, the following objectives should be achieved:

1. To perform literature analysis in the research field to elaborate a research workflow, covering all necessary problem-solving stages.
2. To inspect the AIS data and apply data preprocessing techniques to propose an appropriate scheme for data preparation according to the different nature of AIS data. The schema includes data structuring, cleaning, down-sampling, missing values imputation, feature engineering, the missing vessel type classifier, and splitting to sequences of vessel navigational vectors, with the view to prepare the data for upcoming anomaly detection analysis.
3. To introduce a method that can solve the imputation problem of missing vessel type values in data. To develop and test vessel type classifier to cope with the issue in the real-world AIS data set.
4. To inspect semi-supervised (point-based) methods for anomaly detection, propose enhancement and explore the possibility to use historical vessel movement data to speed up the semi-supervised algorithm while analyzing streaming AIS data.
5. To inspect unsupervised (trajectory-based) methods for anomaly detection, define extraction technique for abnormal vessel movement region, and compare the obtained results using methods based on statistical techniques.
6. To perform a comparative analysis of abnormal trajectories obtained by applying semi-supervised and unsupervised methods on AIS data by investigating a region at two sea areas.

Research Methods

Research that was performed in this thesis is based on these scientific methods:

1. Literature review is performed on the latest scientific papers in the research field to identify, select and evaluate state-of-the-art algorithms for solving the stated problem.
2. Quantitative and qualitative information gathering was performed to create data sets, which were used for experiments and experi-

mental data describing the performance of the proposed solution or its components.

3. Methods including but not limited to statistical ones were used to perform confirmatory data analysis, ensuring the reliability of data and experimental setup.
4. Exploratory Data Analysis (EDA): Box plot, Histogram, Scatter plots, Pair plots, Negative likelihood contour plots.
5. Descriptive Statistics: Univariate Analysis, Multivariate Analysis, Pearson's correlation, Cramer's V correlation; Data mean variance scaling; Synthetic Minority Oversampling Technique (SMOTE).
6. Model evaluation: classification confusion matrix, classification metric comparison, evaluation and comparison of regression errors, Prediction Region Coverage Probability (PICP) and Prediction Region Normalized Average Width (PINAW), Wild bootstrapping techniques.
7. Multivariate clustering techniques: Self-Organizing Map, Soft-DTW k-means.
8. Dimensionality reduction techniques: Multi Dimensional Scaling.
9. Artificial neural network techniques: LSTM; Multi Layer Perceptron (MLP); Auto-encoders; Neural network layers stacking.
10. Constructive research was used to propose improvements to the solution of the real-world problem and propose new methods to improve Maritime Situational Awareness (MSA).
11. Software development and parallel computation methods with GPUs and TPUs were used in the experimental part of this thesis, including the implementation of marine vessel anomaly detection and trajectory clustering.

Scientific Contributions and Practical Value of the Research

This thesis contributes to the development of marine vessel traffic anomaly detection as an extension to Maritime Situational Awareness (MSA). The main contributions of this thesis can be outlined as follows:

1. The point-based modified Self-Organizing Map (SOM) algorithm for marine vessel movement data classification into normal and abnormal classes is proposed and investigated on two independent data sets. The modification is done by incorporating virtual pheromone intensity calculations at the last stage of model training. This method has shown better classification results on less intense (less than 140,000 navigational vectors) marine vessel traf-

fic data sets. The procedure for selecting the best neighbourhood function and SOM grid size is introduced.
From the practical point of view, it can improve Maritime Situational Awareness for Vessel Traffic Service of relatively small ports with moderate traffic.

2. The retraining strategies for SOM point-based methods are proposed. Applying different SOM model retraining strategies while keeping the same data batch sizes substantially decreased the time for retraining the maritime traffic abnormal movement detection model sustains precision and sensitivity at very high values. The results obtained show that the SOM network could be retrained in half the time while keeping precision and sensitivity at almost the same high values.
In practice, it can increase speed and shorten the time for model retraining by keeping the model updated with the most up-to-date data or significantly reduce the cost of hardware required for model training.

3. Vessel type prediction method is proposed for missing vessel type imputation by vessel trajectories using multi-stacked multivariate Long Short Term Memory (LSTM) method. Such classification experiment has shown that classification precision and sensitivity are satisfactory and can be used for this purpose.
In practice, it enriches the training data set with additional training samples.

4. Two LSTM based methods were proposed for unsupervised detection of abnormal marine vessel trajectories. Both methods detect anomalies by checking trajectories in the prediction region. First, the LSTM prediction learning method was created by modification of univariate LSTM interval learning to learn multivariate prediction region. Second, the LSTM wild bootstrapping method based on the integration of statistical wild bootstrapping technique was adapted to LSTM multi-stacked multivariate auto-encoder to create prediction region ellipses for normal movement model. Both methods show the ability to detect a broader range of anomalous trajectory line shapes compared to SOM based methods.
In practice, it could simplify the anomaly detection models' training by avoiding vessel trajectory labelling for anomalous traffic cases, which is usually required for tuning semi-supervised models based on semi-supervised SOM methods. LSTM method could be used for larger areas or sea areas with substantial traffic, where labelling of abnormal trajectories is unfeasible. The wider range of detected anomalous trajectories improve Maritime Situational

Awareness for Vessel Traffic Service.

Defensive Claims

The following claims are defended in this thesis:

1. Proposed SOM neural network with integrated virtual pheromone for detection of vessel traffic anomaly performs better on smaller data sets than Self-Organizing Map (SOM) with integrated Gaussian Mixture Model (GMM) (SOM_GMM). However, the SOM_GMM should be used for the larger sets.
2. SOM neural networks can be retrained for anomaly detection tasks in a shorter time with a minor change in precision compared to classical training workflow.
3. The proposed Long Short Term Memory (LSTM) prediction region learning and LSTM wild bootstrapping methods can detect vessel trajectory anomalies. The LSTM prediction region learning outperforms LSTM wild bootstrapping method on quite small data sets.
4. LSTM architecture with good generalization properties can be applied for the detection of vessel type to perform an imputation of missing values.
5. Point-based anomaly methods Self-Organizing Map (SOM) neural network with integrated virtual pheromone and Self-Organizing Map (SOM) with integrated Gaussian Mixture Model (GMM) do not detect anomalies in trajectories with sharp manoeuvres and stopping line shapes but LSTM methods do.

Approbation of the Results

Results obtained in this thesis were published in 4 papers:
3 papers in periodic scientific journals indexed by Web Of Science and 1 paper in reviewed scientific conference proceedings. The results were presented at 8 international scientific conferences. The following list presents the publications and presentations in conferences:

Papers in periodic scientific journals indexed in The Web of Science:

- J. Venskus, P. Treigys, and J. Markevičiūtė. "Unsupervised Marine Vessel Trajectory Prediction using LSTM Network and Wild Bootstrapping Techniques". Nonlinear Analysis: Modelling and

Control. (2021). Vilnius University. ISSN 1392-5113 | eISSN 2335-8963. (IN PRINT)

- Venskus, Julius; Treigys, Povilas; Bernatavičienė, Jolita; Tamulevičius, Gintautas; Medvedev, Viktor. Real-time maritime traffic anomaly detection based on sensors and history data embedding // Sensors. Basel : MDPI. ISSN 1424-8220. 2019, vol. 19, no. 17, art. no. 3782, p. 1-10. DOI: 10.3390/s19173782.
- Venskus, Julius; Treigys, Povilas; Bernatavičienė, Jolita; Medvedev, Viktor; Voznak, Miroslav; Kurmis, Mindaugas; Bulbenkienė, Violeta. Integration of a self-organizing map and a virtual pheromone for real-time abnormal movement detection in marine traffic // Informatica. Vilnius : Vilniaus universiteto Matematikos ir informatikos institutas. ISSN 0868-4952. 2017, Vol. 28, No. 2, p. 359-374.

Papers in peer-reviewed scientific conference proceedings:

- Venskus, Julius; Treigys, Povilas. Meteorological data influence on missing Vessel type detection using deep Multi-Stacked LSTM neural network // Computer data analysis and modeling: stochastics and data science : proceedings of the XII international conference, Minsk, September 18-22, 2019. Minsk : Belarusian State University, 2019. ISBN 9789855668115. p. 307-310.

Presentations in scientific conferences:

- Venskus, Julius; Treigys, Povilas. Meteorological data influence on missing Vessel type detection using deep Multi-Stacked LSTM neural network // Computer data analysis and modeling: stochastics and data science : proceedings of the XII international conference, Minsk, September 18-22, 2019. Minsk : Belarusian State University, 2019. ISBN 9789855668115. p. 307-310.
- Venskus, Julius; Treigys, Povilas. Preparation of training data by imputation missing vessel type data using deep multi-stacked LSTM neural network for abnormal marine transport evaluation // ITISE 2019 : International Conference on Time Series and Forecasting : proceedings of abstracts. Granada, Spain, September, 25-27, 2019. Granada : Universidad de Granada, 2019. ISBN 9788417970796. p. 38.
- Venskus, Julius; Treigys, Povilas; Bernatavičienė, Jolita; Retraining strategies of modified SOM for abnormal marine traffic detection; Materials, Methods & Technologies 2018 : 20th International

conference. Elenite, Bulgaria, June 26-30, 2018. International Scientific Events.

- Venskus, Julius. Saviorganizuojančių žemėlapių ir virtualių feromonų integravimas jūrinio transporto avarinių realaus laiko judėjimų nustatymui. XVIII mokslinė kompiuterininkų konferencija. Kaunas, 2017 m. rugsėjo 21–22 d.
- Venskus, Julius; Treigys, Povilas; Bernatavičienė, Jolita; Markevičiūtė, Jurgita. Detecting Maritime traffic anomalies with long-short term memory recurrent neural network // 11th international workshop on data analysis methods for software systems (DAMSS 2019), Druskinin-kai, Lithuania, November 28-30, 2019 / Lithuanian Computer Society, Vilnius University Institute of Data Science and Digital Technologies, Lithuanian Academy of Sciences. Vilnius : Vilnius University Press, 2019. ISBN 9786090703243. eISBN 9786090703250. p. 89. DOI: 10.15388/Proceedings.2019.8.
- Venskus, Julius; Treigys, Povilas; Bernatavičienė, Jolita; Andziulis, Arūnas. Aspects of data collection for abnormal marine transport evaluation // DAMSS 2018 : 10th international workshop on "Data analysis methods for software systems", Druskininkai, Lithuania, November 29 - December 1, 2018 : [abstract book]. Vilnius : Vilniaus universitetas, 2018. ISBN 9786090700433. p. 88.
- Venskus, Julius; Treigys, Povilas; Bernatavičienė, Jolita; Medvedev, Viktor. Retraining strategies of modified SOM for abnormal marine traffic detection // 9th International workshop on Data Analysis Methods for Software Systems (DAMSS), Druskininkai, Lithuania, November 30 - December 2, 2017. Vilnius : Vilniaus universitetas, 2017. ISBN 9789986680642. p. 54.
- Venskus, Julius; Kurmis, Mindaugas; Treigys, Povilas. Modified SOM for abnormal marine traffic detection // Data analysis methods for software systems : 8th international workshop on data analysis methods for software systems, Druskininkai, December 1-3, 2016. Vilnius : Vilniaus universiteto leidykla, 2016. ISBN 9789986680611. p. 66-67.

Presentations in national scientific institutions:

- Venskus, Julius. Unsupervised Marine Vessel Trajectory Prediction using LSTM Network and Wild Bootstrapping Techniques. Klaipeda University. Department of statistics and computer science. 2020m. September.
- Venskus, Julius. Investigation of Unsupervised Machine Learning Methods for Detection of Sea Traffic Anomaly. Lithuanian Mar-

itime Academy. 2021.

- Venskus, Julius. Meteorologinių duomenų įtaka nustatyti trūkstamai informacijai apie jūrų laivo tipą, naudojant daugiasluoksnius LSTM neuroninius tinklus. System analysis seminar. Vilnius University. Institute of Data Science and Digital Technologies. 2019 October 7.
- Venskus, Julius. Jūrų laivo tipo atpažinimas pagal eismo duomenų seką naudojant daugiasluoksnį LSTM tinklą. System analysis seminar. Vilnius University. Institute of Data Science and Digital Technologies., 2019m. February. Vilnius
- Venskus, Julius. Savi-organizuojančio neuroninio tinklo (SOM) ir virtualaus feromono tyrimas neįprastam laivų eismo aptikimui. System analysis seminar. Vilnius University. Institute of Data Science and Digital Technologies. 2018 February. Vilnius
- Venskus Julius, Savi-organizuojančio neuroninio tinklo (SOM) ir virtualaus feromono integravimas neįprastam laivų eismo aptikimui koncepcijos pristatymas. System analysis seminar. Vilnius University. Institute of Data Science and Digital Technologies. 2017m. June 4th. Vilnius

Outline of the Thesis

The research schema depicted in Figure 1 and numbered steps are noted below in the outline of the thesis, which is organized as follows:

- Introduction section provides an introduction to the research and overview of the dissertation.
- Section 1 reviews related work in the same research area including detection of abnormal marine vessel movements and prediction of spatio-temporal sequence (step 1).
- Section 2 introduces description of data sources (steps 2, 3, 4, 5), structure of data, restructuring of data, cleaning of raw data, downsampling, imputation of general case missing values, feature engineering, splitting to sequences, and imputation of missing vessel type data (steps 6, 7, 8). This section presents a method of vessel type classification by historical trajectory and results of trained model testing (step 6).
- Section 3 presents a design of point-based semisupervised marine vessel traffic anomaly detection (step 9) based on SOM and virtual pheromone integration (step 11), and SOM with Gaussian Mixture Model (GMM) (step 12). This section describes details about these methods, parameter selection, and retraining strategies (step 9).

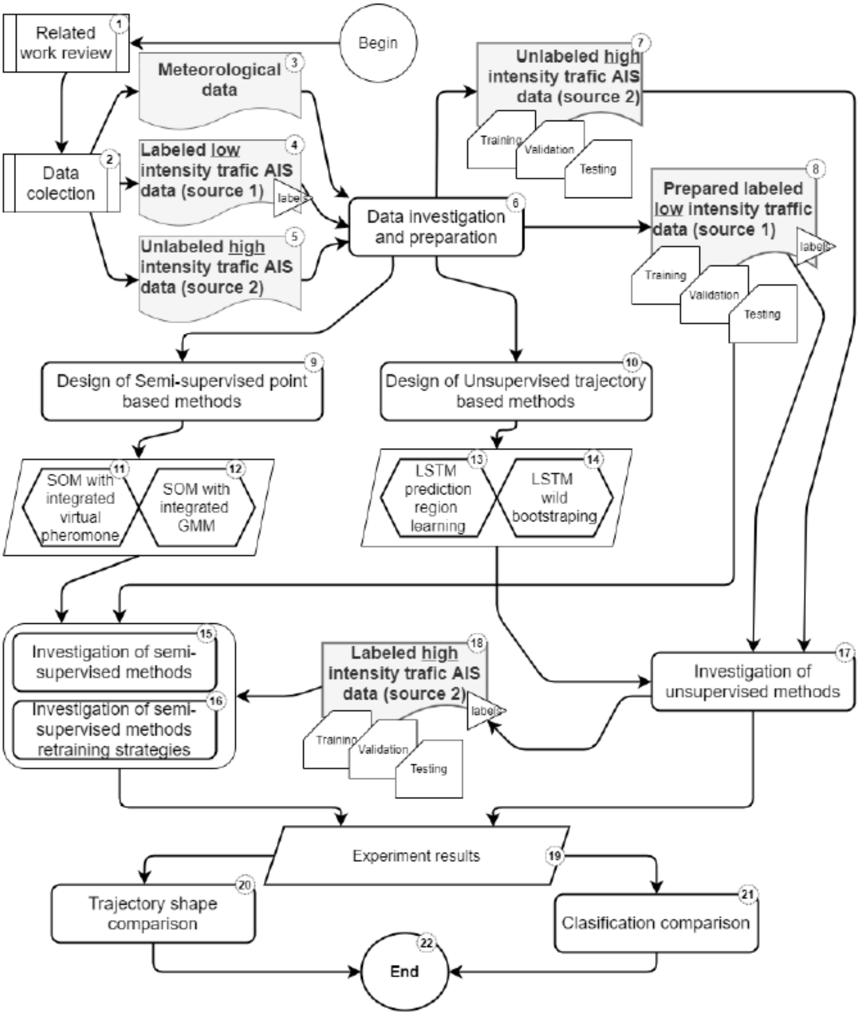

Begin
Related work review
1
Data colection
2
Meteorological data
3
Labeled low intensity trafic AIS data (source 1)
4
label
Unlabeled high intensity trafic AIS data (source 2)
5
Data investigation and preparation
6
Unlabeled high intensity trafic AIS data (source 2)
7
Training
Validation
Testing
Prepared labeled low intensity traffic data (source 1)
8
labels
Training
Validation
Testing
Design of Semi-supervised point based methods
9
Design of Unsupervised trajectory based methods
10
SOM with integrated virtual pheromone
11
SOM with integrated GMM
12
LSTM prediction region learning
13
LSTM wild bootstraping
14
Investigation of semi-supervised methods
15
Investigation of semi-supervised methods retraining strategies
16
Investigation of unsupervised methods
17
Labeled high intensity trafic AIS data (source 2)
18
labels
Training
Validation
Testing
Experiment results
19
Trajectory shape comparison
20
Clasification comparison
21
End
22

Figure 1: Research schema

- Section 4 describes the design of the proposed algorithms and methods for unsupervised detection (step 10) of marine vessel abnormal trajectories. The design of the two methods is described. These methods are based on LSTM prediction region learning (step 13) and LSTM wild bootstrapping (step 14).
- Section 5 contains the description of experiments and the results related to both point-based semisupervised methods (steps 15, 16) and trajectory-based unsupervised methods (step 17). The LSTM methods are investigated (step 17) using both data sources (from steps 8, 7) and the results collected (step 19). The SOM_GMM and SOM_pheromone methods are investigated using labelled data

sets from the first source (step 8) and data sets labelled using LSTM methods from the second source (step 18). The point-based method's retraining strategies were investigated (step 16), and the results were stored (step 19). The classification metrics of all methods were compared (step 21) and trajectories of anomalous traffic were compared (step 20).

General conclusions are presented after Section 5; 93 bibliographic references are included at the end of the thesis.
The dissertation consists of 105 pages, 28 figures, 34 tables, and 7 appendixes.

1 Related Work

1.1 Detection of Abnormal Marine Vessel Movement

International Maritime Organization Regulation 19 of SOLAS [20] Chapter V created in 2000 and later in 2015, they revised a requirement for all vessels to carry automatic identification systems (AISs) capable of providing information about the marine vessel to other vessels and coastal authorities automatically [21]. Thus AIS is able to gather dynamic and static vessel traffic data. The vessel traffic anomaly detection can be defined as a task in AIS data analysis and outlier detection, where vessel traffic data are analyzed as multiple standalone vessels positions/navigational vectors (point-based) or in a trajectory-based manner where vessel's vectors are structured to time series sequences [11].

Loi *et al.* [22] propose that historical vessel vector data can be mined by cell-based method. The authors divide the region of interest to subareas (cells) of predefined size. In each cell, the probability density for the speed data is estimated based on a normal kernel function. Then the DBSCAN algorithm is used to create speed clusters of normal moving speeds. Speed that is out of clusters is defined as abnormal. Ristic [23] presents an unsupervised method that subdivides geographical area into individual cells and AIS data with the same coordinates are assigned to these cells. The vessel navigational data in the grid are analyzed using signature, rule and Poisson point process based techniques in order to detect various association rules in movement changes. Zhu [24] applied data mining technologies to analyze AIS data in data warehouses. Arguedas [12] proposed to automatically produce synthetic maritime traffic representations from historical self-reporting self-positioning systems or meteorological and oceanographic positioning data. S.K.Singh and F.Heymann [25] presented a multi-class artificial neural network (ANN). The multi-class anomaly framework captures AIS message dropouts, channel effects or intentional messages potentially related to illegal activities. The author uses three layer ANN and data structured to time series sequences per vessel. The model was trained in a supervised manner. Venskus *et al.* [26] used a Self-Organizing Map (SOM) combined with virtual pheromone for anomaly detection. Later retrain techniques were proposed [27]. Pallota *et al.* [28] propose incremental and unsupervised point-based analysis of traffic anomaly. Despite of authors' declaration that method is point-based, they updated their model based on historical traffic knowledge. The sliding time window technique was used to find the relations between successive vessel's navigational vectors. The discovered way points were clustered using DBSCAN method and it was used for abnormal traffic detection [29, 30]. The authors in paper [27] state that the main weakness of point-based techniques is that the analysis of movement is based on short-term history or it even disregards the history. On the other hand, a limited number of analyzed data points intend real-time calculation and decision making. This quality makes point-based anomaly detection techniques attractive for real-time tasks. Nevertheless, at the moment, the prevalence of these techniques is quite limited. The majority of vessel traffic techniques are based on trajectory analysis where vessels vector data are analysed as sequences in time. In the literature, several research directions can be found such as risk assessment of vessel collision [31], vessel traffic anomaly detection [32, 33, 34] vessel type identification [35, 34], and fault detection [36]. Trajectory-based models of marine normal traffic/movement are created based on vectors on an entire trajectory. The abnormal motion is detected when an analyzed vessel trajectory has a deviation from a model. These techniques demand a vast amount of AIS data for analysis. However, it helps to create models that take into account traffic history and detect more complex trajectory-based anomalies. The rise of parallel computational power enables faster processing of a considerable amount of data. Graphical processing units (GPU) and Tensor processing units (TPU) play a big part.

Data preparation techniques give a significant boost in performance for these types of models. Tang *et al.* [37] propose a detection of abnormal vessel behaviour by applying directed graph model. The model has three modules: data processing, model construction, and abnormal behaviour detection module. The data pre-processing greatly affects the efficiency. The author converted each trajectory to a mesh grid representation. Based on grid representation, a directed graph is created. The statistical characteristics are obtained by analysing the course and speed distribution of the vessels in each node in the directed graph. When the monitored vessel's trajectory is outside of confidence level 99.7%, the vessel trajectory is considered anomalous. Lu *et al.* [17] briefly reviewed shape-based vessel trajectory similarity and clustering. Authors review shape-based similarity computing methods: Hausdorff distance, Frechet distance, and SSPD distance. In the same paper, unsupervised algorithms for trajectory clustering were experimentally evaluated using spectral clustering, hierarchical cluster analysis with distances average linkage, complete linkage, and ward linkage.

However, when vessel traffic data is analysed as trajectories (each vessel's navigation location sequences), the model takes advantage of historical vessel behaviour information. Historical vessel behaviour is crucial for its anomaly detection and such type of analysis requires that more complicated methods are used. Recent advances in deep learning techniques provide a possibility to train model complicated nonlinear patterns using a large amount of data.

To detect marine traffic anomalies, the authors use two main approaches. In the first approach, the analyzed vessel trajectory similarity is compared against model trajectories or clusters of trajectories [17] [32]. The compared similarity distance is tested against a linear threshold or normal distribution of trajectories in a cluster. In the second approach, the trained model uses the vessel trajectory to predict the following trajectory points. Then the predicted trajectory is compared with the actual vessel trajectory [37], and if the actual trajectory is outside of the predefined confidence/prediction level, it is classified as abnormal. The first approach should capture complex non-linear patterns and be mostly used in arbitrary non-intensive marine traffic. With the second approach, the authors could reach better results by capturing intricate patterns in intensive marine traffic regions. Despite that, the second approach can be used with deep learning regression techniques for vessel trajectory prediction with prediction level evaluation.

From the other perspective, as the vessel sends the data to the AIS system with time dynamics and shows a change in the vessel location in space, other authors interpret vessel trajectories as a task of spatio-temporal data analysis. Classical machine learning methods hardly capture complex spatio-temporal patterns in heavy marine traffic and a large amount of data. The deep learning approach must be investigated to predict and evaluate prediction intervals to detect complex trajectory anomalies in maritime traffic.

1.2 Spatio-temporal Sequence Prediction

The transport traffic flow or trajectory prediction can be analysed as spatio-temporal system, where the data represents the space and the time relation. In this subsection, we will briefly review machine learning methods used for spatio-temporal sequence forecasting. Shi *et al.* [38] classify the spatio-temporal sequence forecasting problems into three categories based on characteristics of the coordinate sequence and measurement sequence, where coordinates stand for vessel geographical location, and measurements stand for navigational variables of a vessel such as speed, heading, type, etc.:

- The first category is Spatio-Temporal Forecasting on Regular Grid (STSF-RG). It has fixed coordinates on a regular grid and prediction is performed for measurements. This category covers problems like video feed [39], crowd density [40], precipitation forecasting/predictions [41].

- The second category has fixed coordinates as well, but the coordinates are on an irregular grid and the category is called Spatio-Temporal Forecasting on Irregular Grid (STSF-IG) [38]. Algorithms of the category are applicable for ground traffic speed prediction [42], where speed measurement stations are sparsely distributed across a city. It can be used for influenza prediction [43], air quality forecasting [44], taxi demand prognosis. [45].

- The third category is The Trajectory Forecasting of Moving Point Cloud (TF-MPC). This category can be characterized by changing coordinates and fixed/changing measurements. The category applies to problems such as human movement trajectory and dynamics prediction [46, 47, 48, 49].

Trajectory Forecasting of Moving Point Cloud (TF-MPC) category can cover not only moving people as entities in moving point cloud. It can be any general moving objects and especially moving marine vessels because they have changing coordinates and measures (speed, heading, etc.) that change in time. A few techniques are used to predict location as well as other measurement values from the data in this category. Most of the techniques are observed based on classical machine learning [45, 50, 43, 47]. As it was discussed in the previous paragraph, the classical methods hardly cope with complex non-linear patterns in big data and therefore the deep learning approach must be investigated. The latest research publications mainly focus on deep learning to harvest the full potential of big data and the possibility to learn complex patterns [51, 40, 48, 5]. Long short term memory (LSTM) neural network, an improved version of recurrent network (RNN), takes advantage of historical data to train deep neural networks. One of them is the Vanilla LSTM [7], which has difficulty in capturing spatial features. To overcome this issue, the convolution layer was introduced to the architecture of LSTM network [41, 51]. He *et al.* [5] proposed Spatio-Temporal Neural Network (STNN) and Li *et al.* [6] introduced LSTM auto-encoder (LSTM-AE) with similar properties, which improved region-based prediction of spatio-temporal data by a smoothing regularization term that was added into the combined model, leading to a more stable estimation.

Despite advances in the prediction of spatio-temporal data with deep neural networks, the authors do not propose an evaluation of prediction or confidence interval, which is crucial for marine traffic anomaly detection with this method. Cruz *et al.* [52] has proposed a univariate solution for estimation of LSTM prediction interval by joint supervision, but this approach is not sufficient because marine traffic has multivariate time series and therefore the approach must be improved.

1.3 Conclusions of the Section

Background information related to research in Detection of Abnormal Marine Vessel Movement and Spatio-temporal Sequence Prediction was provided in this section.

In researches, the vessel traffic data are analyzed as multiple standalone vessel positions/navigational vectors (point-based) or in a trajectory-based manner where vessel vectors are structured to time series sequences. Data preparation techniques give a significant boost in performance for these types of models. Historical vessel behaviour is crucial for its anomaly detection, and such type of analysis requires that more complicated methods are used.

Related research shows that a vessel sends the data to the AIS system with time dynamics and shows a change in the vessel location in space, and vessel trajectories can be interpreted as a task of spatio-temporal data analysis. Classical machine learning methods hardly capture complex Spatio-temporal patterns in heavy marine traffic and diverse data. The deep learning approach must be investigated to predict and evaluate prediction intervals to detect complex trajectory anomalies in maritime traffic.

Semi-supervised methods require anomaly-labeled training data sets. In practice, it is not feasible to perform labelling of enormous size data sets. Unsupervised techniques must be used. The unsupervised LSTM artificial neural networks with prediction regions can be a promising technique to perform marine traffic trajectory anomaly detection.

2 Data Preparation

This section contains a description of data sources, data structure, restructuring of data, raw data cleaning, down-sampling, imputation of general case missing values, feature engineering, splitting to sequences, and missing vessel type imputation. These steps are needed to ensure equal conditions for the investigative needs. Final data sets of marine vessel traffic are described and prepared at the end of this section.

2.1 Description of the Data Sources

Three data sources were used in experiments. The first source contains data of marine traffic, which was obtained from Automatic Identification System (AIS) and collected/stored by the Danish maritime authority [53]. The second data set represents meteorological data set that was obtained from European Centre for Medium-Range Weather Forecasts (ECMWF) through World Weather Online service API [54]. And the third data source contains data about Klaipeda seaport region (AIS) and was taken from Klaipeda Sea Port authority [26, 27].

For the final study of the anomaly detection algorithms under conditions of high intensity marine traffic, the choice of sea area "Fehmarnbelt" in the Baltic Sea was based on the data collected by Danish maritime authority. This area is well known for its very high intensity region in Danish and German waters. There is a main intersection area of marine routes from/to countries such as Russia, Poland, Sweden, Finland, Germany, Lithuania, Estonia, Latvia, and Denmark. This area has several large ports such as Kiel, Lubeck, Wismar, and Rostock. In order to keep high control of maritime awareness and minimise security risk, the Vessel Traffic Service (VTS) in this region requires automated solutions to help operators to make timely decisions.

AIS data set. The AIS historical database for this research contains data on historical AIS maritime vessel traffic in Danish waters. The whole database has data from 2006 to 2020. Multiple records of this data set contain AIS navigational vectors of multiple vessels. A single vector contains navigational parameters/properties, which are listed in the Tables 1, 2, 3. The AIS data records consist of three field categories. The first category (see Table 1) has fields of static data. This category represents logical and physical properties of a single vessel and it does not change within the same vessel data. The fields in the second category are shown in Table 2. Dynamic category contains data from vessels' onboard sensors and they change dynamically during the whole voyage. Table 3 contains data fields of voyage categories. Those fields contain

Table 1: Description of static AIS record fields

Field Name	Data type	Description
Type of mobile	String	Describes what type of target this message is received from
MMSI	Long	Maritime Mobile Service Identity (MMSI) number of vessel
IMO	Long	Vessel identifier provided by International Maritime Organization (IMO)
Callsign	String	Call sign of the vessel
Name	String	Name of the vessel
Ship type	String	Describes the AIS vessel type of this vessel. The field has limited number of string variations
Width	Integer	Width of the vessel
Length	Integer	Length of the vessel
Type of position fixing device	String	Type of position fixing device from the AIS message
Draught	Integer	Draught field from AIS message
Data source type	String	Type of data source, e.g. AIS

information about a single voyage of a particular vessel.

From the whole Danish AIS historical database, a single geographical region "Fehmarnbelt" was filtered out as described above in this subsection and details are summarized in Table 5.

Table 2: Description of dynamic AIS record fields

Field Name	Data type	Description
Timestamp	Date & Time	Timestamp from the AIS base station
Latitude	Decimal	Latitude of message report (e.g. 57.8794)
Longitude	Decimal	Longitude of message report (e.g. 17.9125)
Navigational status	String	Navigational status from AIS message if available, e.g.: 'Engaged in fishing', 'Under way using engine', etc.
ROT	Decimal	Rot of turn from AIS message if available
SOG	Integer	Speed over ground from AIS message if available
COG	Integer	Course over ground from AIS message if available
Heading	Integer	Heading from AIS message if available

Meteorological data set. Meteorological data set was obtained using World Weather Online service API [54] in the European Centre for Medium-Range Weather Forecasts (ECMWF) grid. This data contains information about wind direction, wind strength, swell direction, swell height, swell period, day/night, and tide level. Meteorological data are

Table 3: AIS record fields of voyage category

Field Name	Data type	Description
Cargo type	String	Type of cargo from AIS message
Destination	String	Destination from AIS message
ETA	String	Estimated Time of Arrival, if available

provided periodically in 3 hour periods. The data were collected from November 1, 2019 to June 31, 2020. A detailed description is presented in Table 4.

Table 4: Record fields of meteorological data

Field Name	Data type	Description
datetime	Datetime	Date and time of meteorological record
latitude	Decimal	The latitude of location (e.g. 57.8794)
longitude	Decimal	The longitude of location (e.g. 57.8794)
winddirDegree	Integer	Wind direction in degrees
windspeedMeterSec	Decimal	Wind speed in meters per second
swellheight	Decimal	Swell height at location
swelldirection	Integer	Swell direction at location
swellperiod	Decimal	Swell period at location
day/nigh	String	String indicates day or night is requested information for specific location

2.2 Data Structuring

"Fehmarnbelt" geographical sea region was chosen in order to study performance and properties of marine anomaly detection methods in the complicated marine region with intense traffic. The Table 5 contains general description of data sets about the geographical region.

The source of the vessel traffic data is AIS. An automated vessel monitoring system is used to monitor and track vessel traffic. Each marine vessel is equipped with an AIS system transponder that sends navigational information from vessels at predefined time intervals. Navigational information includes data shown in Tables 1, 2, 3. The obtained data from AIS is stored in databases and is used by Vessel Traffic Service (VTS) to monitor and control vessel traffic. AIS plays important role in a Maritime Situational Awareness (MSA).

The raw data is stored in a flat structure, where each record consists of vessel's navigational data at a certain time. The data structure can be represented by:

Table 5: General information of data set

Geographical region name	Fehmarnbelt
Time period	from 2019-01-01 00:00:00 to 2019-03-31 23:59:59
Latitude interval, degrees	53.832833 to 54.998114
Longitude interval, degrees	9.97929 to 12.53451
Memory usage in mega bytes of AIS data	73579.26
N' - total number of AIS records (1)	98245370
f - initial* number of AIS fields/features (1)	11
V - total number of unique vessels by MMSI (2)	3913
Meteorological grid size	$4 \times 7 \times 0.5°$
Meteorological data gathering locations	28
Total number of meteorological vectors	20608
Memory usage in mega bytes of meteorological data	822.05

* - the initial fields/features list contains features from the original AIS data sets before preprocessing of data set. After preprocessing the AIS data set may contain a different number of features.

$$
\begin{gathered}
X = \{x_1, x_2, \ldots, x_g, \ldots, x_{N'-1}, x_{N'}\}, \\
x_g = \left[x_g^{(1)}, x_g^{(2)}, \ldots, x_g^{(j)}, \ldots, x_g^{(f-1)}, x_g^{(f)}\right], \qquad (1) \\
g \in \{1, 2, \ldots, N'\}, \quad j \in \{1, 2, \ldots, f\},
\end{gathered}
$$

where x_g is a single vessel navigational data record consisting of $x_g^{(j)}$ fields, that are unique vessel identifier MMSI, 'Latitude', 'Longitude', SOG, COG, Ship type, Timestamp of the data being received. The whole data set contains N' number of records, where each record consists of f fields/features of vessel navigational vector. Accordingly, the indexes g and j denote a particular vector ordered location in the data set and feature list. The data set X contains multiple vessel navigational unordered vectors.

Vessels send navigational data periodically thus received data instance is stored in the order the data was received (ordered by the timestamp). The raw data that is structured in this way is difficult to work with multiple navigational data of different vessels, which forms vessel's sea path over time. In order to investigate models that are discussed more thoroughly in section 5, one needs to train models to predict a single vessel path. To achieve that, we restructure data structure presented by equation (1) per unique vessel, vessel's navigational data is grouped by unique vessel identifier MMSI and ordered by the timestamp. The restructured data set can be presented as:

$$S = \left\{ \begin{array}{c} s_1 \\ s_2 \\ \vdots \\ s_v \\ \vdots \\ s_V \end{array} \right\} = \left\{ \begin{array}{c} \left\{ x_{(1,1)}, x_{(1,2)}, \ldots, x_{(1,p_1)}, \ldots, x_{(1,P_1)} \right\} \\ \left\{ x_{(2,1)}, x_{(2,2)}, \ldots, x_{(1,p_2)}, \ldots, x_{(2,P_2)} \right\} \\ \vdots \\ \left\{ x_{(v,1)}, x_{(v,2)}, \ldots, x_{(v,p_v)}, \ldots, x_{(v,P_v)} \right\} \\ \vdots \\ \left\{ x_{(V,1)}, x_{(V,2)}, \ldots, x_{(V,P_V-1)}, \ldots, x_{(V,P_V)} \right\} \end{array} \right\}, \tag{2}$$

$$v \in \{1, 2, \ldots, V\}, \; p_v \in \{1, 2, \ldots, P_v\}, \quad P_v \in \{P_1, P_2, \ldots, P_V\}$$
$$x_{(v,p_v)} \in s_v \subset S \subseteq X, \quad \forall x (x \in S \rightarrow x \in X)$$

where S is the restructured vessel navigational vector data set, obtained by transforming the X (see eq. (1)). Each row of the matrix consists of a set of navigational data vectors s_v of an individual vessel's navigational vectors, where v denotes the index of a distinct vessel's navigational vectors set and V denotes the number of different vessels. p_v denotes the index of an individual vector from single vessel vectors set S_v. These navigational data sets for each vessel are made of the navigational vectors x_g reindexed to $x_{(v,p_v)}$ by following rules:

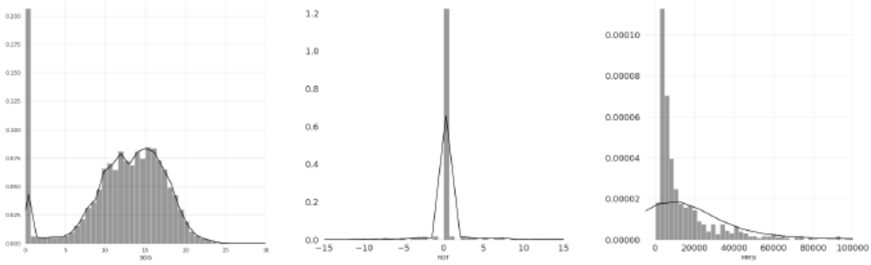
- All vectors from X data set are grouped into subsets $\{s_1, s_2 \ldots, s_v, \ldots, s_V\}$. The grouping is done by vessel identifier field MMSI that is part of features and can be noted $x^{(MMSI)}$. After this operation, each s_v contains vectors of individual vessel navigational. The quantity of each s_v vector is different and noted P_v. The number V of unique vessels can be found in Table 5.

- The vector subsets s_v of each vessel are sorted sorted by the data acquisition time field/feature. This feature is denoted by $x^{(timestamp)}$. The sorting is performed to match inequality $x_{(v,p_v-1)}^{(timestamp)} < x_{(v,p_v)}^{(timestamp)}$. The vectors that were received earlier in time have lower index number.

Each row in S matrix has a different number P_v of vectors x_{v,p_v} per vessel and the distribution of these vectors per vessel is depicted in Figure 2c. The distribution shows that most of the vessels have P_v number less than 20,000 vectors. It shows that these vessels are passing, arriving, or leaving the investigated sea region. Figures 2a and 2b will be described in more detail later.

2.3 Data Set Cleaning

AIS systems interconnect many parts such as vessels, Vessel Traffic Service (VTS), Receiving base stations, and collection databases. The Dan-



(a) SOG distribution (b) ROT (c) Vectors per vessel

Figure 2: SOG, ROT and vectors per vessel type distributions

ish maritime agency has collected more than 1TB and 10^9 vessel navigational vectors in territorial waters during 2019 [53]. The problem is that enormous quantity of vessels, for example, can have transponders manufacturer by different vendors communicating with AIS through VHF radio band that is sensitive to external noise and is prone to interference with data transmissions from other vessels. These imperfections affect the quality of data that is sent/received. The main discrepancies are of several types. Due to the nature of radio transmission, the data is received from non-uniformly distributed areas. AIS signals can be distorted by the atmosphere and not received properly. Such distortion can result in very short frequent sequences of the same vessel's vector with missing features to be received, and signal duplication. In some cases observed, a vessel sends an incorrect vessel's MMSI identification number and malfunctioning vessel devices or faulty installation of equipment causes wrong data being sent to the AIS station. While vessel is anchored it still sends data that cannot be treated as abnormal traffic, or after vessels leave of the region of interest, radio coverage can be disturbed, which may result in loss of navigational data. Such forms of information losses cause vessel path gaps in analysed vessel trajectories.

To overcome these problems the gathered vessel data from the AIS needs to be cleaned. All vectors that are outside of analyzed marine area must be dropped, all duplicated data, and anchored vessel navigational vectors should be removed. Short sequences of vessel vectors (paths) shorter than predefined training sequence length n' are removed as well.

AIS data cleaning. In order to perform data cleaning the descriptive statistics are calculated. Descriptive statistics of vessels traffic AIS numerical data are shown in the Table 6. Here one can see values for each field, including illegal values. Other statistical properties such as mean, standard deviation, min, max, and percentiles. One can see that

Table 6: Descriptive statistics of vessels traffic AIS numerical data

Field Name	count, 10^6	mean	std	min	max
Latitude	98.2	54.61627	0.23228	53.86743	54.99811
Longitude	98.2	11.22920	0.72786	9.979292	12.53451
ROT	74.8	-0.019	7.782	-0.071	7.080
SOG	96.8	7.19	6.70	0.00	140.00
COG	93.2	176.2	106.6	0.0	359.9
Heading	84.7	174.9	100.4	0.0	510.0
Width	92.8	15.18	9.67	2.00	124.00
Length	92.8	88.63	69.56	3.00	812.00
Draught	87.2	4.725	2.837	0.100	25.50

⋮

Field	Percentiles		
Name	25%	50%	75%
Latitude	54.44638	54.57917	54.83344
Longitude	10.67159	11.22875	11.91921
ROT	0.000	0.000	0.000
SOG	0.00	7.90	12.30
COG	79.3	196.0	268.0
Heading	86.0	190.0	262.02
Width	6.00	13.00	23.0
Length	24.00	80.00	142.0
Draught	2.200	4.20	6.200

all numerical features are in different ranges in the table . That implies the requirement of scaling and normalization prior to model training. In the dissertation, the Min/Max scaling is used to prepare for both types of SOM and LSTM neural networks. This type of scaling is mandatory for LSTM networks because TANH activation function is used.

More statistics are calculated in order to analyse numerical data. A pair plot of numerical AIS data is depicted in the Appendix A. The correlation matrices of vessels traffic AIS numerical and categorical data are shown in Tables 7, 9, and 6. Based on those three tables and domain knowledge, feature selection and data reprocessing decisions are made. The analyzed features are:

Latitude and Longitude - the pair of features represents spatial information about vessel location in World Geodetic System 1984 (WGS84). Despite the fact that deep neural networks are good in adopting to non-linear space, a decision was made to convert spatial data to Euclidean space. By performing WGS84 projection to Universal Transverse Mercator (UTM) coordinate system the vessels spatial data is converted to Euclidean space. Longitude and latitude have no strong correlations (see Table 7) with other features. The geographical distribution of vessel vec-

Table 7: Pearson's correlation matrix of vessels traffic AIS numerical data

	Latitude	Longitude	ROT	SOG	COG	Heading	Width	Length	Draught
Latitude	1.000000	-0.192795	0.001785	-0.248329	0.012123	0.005202	-0.256241	-0.301994	-0.232486
Longitude	-0.192795	1.000000	0.001541	0.234240	0.002224	-0.003220	0.225848	0.246817	0.159610
ROT	0.001785	0.001541	1.000000	0.000528	0.000218	0.002442	0.000449	0.000146	0.001344
SOG	-0.248329	0.234240	0.000528	1.000000	-0.094312	-0.085273	0.561804	0.615442	0.479451
COG	0.012123	0.002224	0.000218	-0.094312	1.000000	0.957538	-0.056304	-0.060491	0.008306
Heading	0.005202	-0.003220	0.002442	-0.085273	0.957538	1.000000	-0.041382	-0.043415	0.022329
Width	-0.256241	0.225848	0.000449	0.561804	-0.056304	-0.041382	1.000000	0.962852	0.818584
Length	-0.301994	0.246817	0.000146	0.615442	-0.060491	-0.043415	0.962852	1.000000	0.827966
Draught	-0.232486	0.159610	0.001344	0.479451	0.008306	0.022329	0.818584	0.827966	1.000000

tors is clearly seen in pair-plot figure (Appendix A).

ROT - it is shown in Table 6. We may note that total count of SOG has 74.8×10^6 values of total 98.2×10^6 values count. In the same table we observe that 75% percentile has value of 0. We see that the majority of ROT has 0 value (Figure 2b). Total count of non zero values is 11.968×10^6. That is 12% of total vectors. Also, ROT is missing in 25% of records and 986 of 3774 vessels lack this feature (see Table 10). Because of that, the decision was made to drop it from the final feature list, which was further investigated.

SOG - by observing statistical parameters of this feature it was noticed it contained a large number of zeros, that is 25% of all vectors. That is clearly visible in figures in Appendix A, 2a and Table 6. Correlation table 7 reveals that SOG shows strong correlation to vessel length, width and moderate correlation to a degree of draught. Note that anomalous behaviour is investigated in moving vessels only, and all anchored vessel records with SOG = 0 have been removed.

COG and Heading - COG feature has 0.4% of total missing values (see Table 10) and shows very high correlation with Heading, (see figures of the Appendix A) and see Pearson's correlation of 0.957538 (depicted in Table 7). Because of very high degree correlation with Heading a decision was made to keep only one feature that has less missing values. 2.5% of values are missing in the Heading. The decision was made in favor of COG.

Table 8: Descriptive statistics of vessels traffic AIS categorical data

Field name	Count, 10^6	Unique values	top category	
			Value	Freq, 10^6
Type of mobile	98.2	7	Class A	95.1
Navigational status	98.2	16	Under way using engine	78.1
IMO	98.2	2963	Unknown	29.3
Callsign	96.1	3756	OX3110	1.3
Name	96.3	4020	Danpilot Echo	1.3
Vessel type	98.2	26	Cargo	26.5
Cargo type	24.6	6	No additional information	14.3
Type of position fixing device	98.2	8	GPS	84.1
Destination	84.1	3484	Rostock	2.8
Data source type	98.2	1	AIS	98.2
MMSI	98.2	3913	219023834	1.3

Width, Length, Draught - These features are physical properties of vessels, that do not change over time. In Figure in Appendix A and Table 7, a strong correlation is seen among all those three features (Width, Length, Draught). Because of high importance of Draught in the maritime domain, it was decided to keep this feature despite the fact it does not change in a single vessel voyage. Because these three parameters are strongly correlated (see Table 7), the Multi Dimensional Scaling (MDS) [55] algorithm was applied in order to keep vessel size influence to the model. The new feature is called "Vessel size". This new feature is used for further model training.

The Table 8 contains description of vessels traffic AIS categorical data. The following properties are analysed: count of values, unique values, top category, frequency of this category. The Table 9 is Cramer's V correlation matrix of vessels traffic AIS categorical data. This analysis is used to overview categorical properties and based on that, a decision was made whether to include or not into a final list of features, which is used to train anomaly detection models. The overview of categorical features is provided below:

MMSI, Name, Callsign, IMO - These features have a strong correlation (see Table 9) with Callsign and Name. The MMSI represents vessel

Table 9: Cramer's V correlation matrix of vessels traffic AIS categorical data

	Navigational status	Callsign	Name	Vessel type	Cargo type	Type of position fixing device	Destination	MMSI
Navigational status	1.000	0.623	0.623	0.261	0.088	0.488	0.596	0.687
Callsign	0.623	1.000	0.998	1.000	0.950	0.994	0.751	0.999
Name	0.623	0.998	1.000	1.000	0.950	0.994	0.754	0.999
Vessel type	0.261	1.000	1.000	1.000	0.266	0.153	0.909	1.000
Cargo type	0.088	0.950	0.950	0.266	1.000	0.143	0.837	0.951
Type of position fixing device	0.488	0.994	0.994	0.153	0.143	1.000	0.778	0.999
Destination	0.596	0.751	0.754	0.909	0.837	0.778	1.000	0.755
MMSI	0.687	0.999	0.999	1.000	0.951	0.999	0.755	1.00

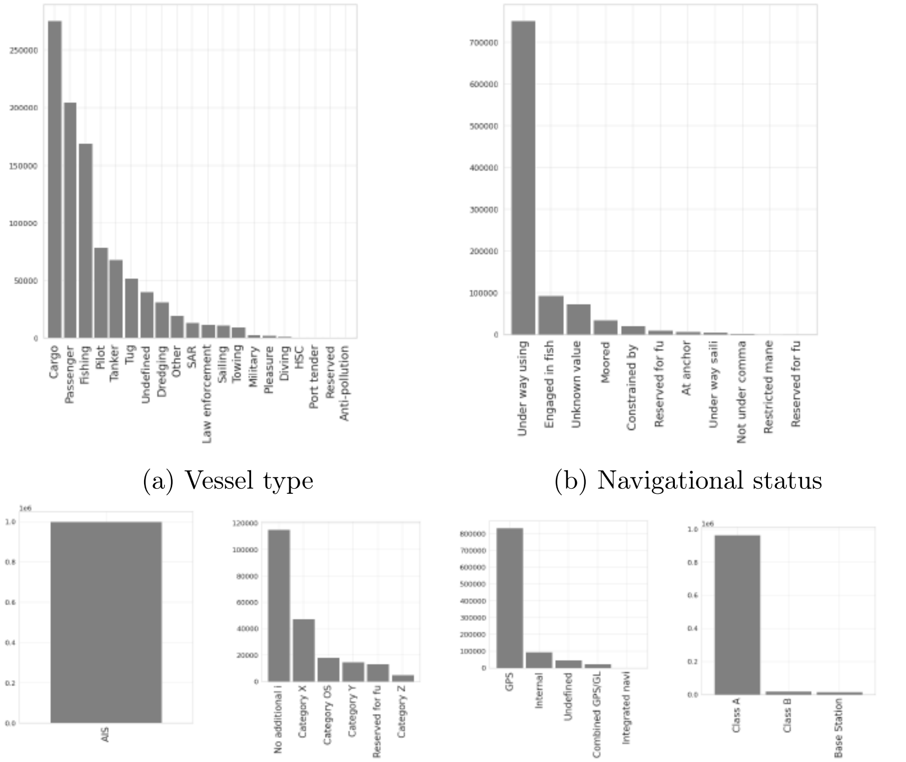
AIS identification number that belongs to single vessel entity. A similar situation is observed with Callsign and Name. These features represent the same vessel. Because of that, it was decided to keep only MMSI and drop Name and Callsign. Moreover, the MMSI is used to structure vectors by vessel (see Equation (2)) in order to form trajectories and spatio-temporal data.

Type of mobile - The majority (97%) of feature values contain string "Class A" (see Table 8 and Figure 3f). Because of that the feature has not been included in further investigation.

Navigational status - The majority of values for this category feature belong to "Under way using engine" category, which contains 79.5% of all values of this feature (see Figure 3b and Table 8). Also, 6% of values for this feature are missing (see Table 10). Because of these reasons and high dissemblance of categories, it was decided to exclude this feature from further creation of models.

Vessel type - The distribution of the feature values is shown in Figure 3a. The feature plays an important role in marine traffic as each vessel type has different behaviour patterns at sea, that are clearly observable in visualisations of vessel traffic by type (see Appendix B) and it has been mentioned in multiple research papers in the same field [56, 14, 26, 28]. Figure 6 shows that each vessel type has different spatio-temporal patterns. Thus, it was decided to split marine traffic data based on vessel type because different vessel types have different behaviour patterns (described in further sub-chapters).

It is important to note that data lack a significant number of missing values regarding vessel type (see Table 10). The vessel type classifier was



(a) Vessel type

(b) Navigational status

(c) Data source type

(d) Cargo type

(e) Fixing device

(f) Type of mobile

Figure 3: Distributions of categorical features

built in order to perform imputation. A more detailed description of the classifier is presented in section 2.8.

Cargo type - The feature value is absent in almost of 85% of AIS vectors and 92% of them do not have this feature at all (see Table 10), therefore the feature was removed from further processing.

Type of position fixing device - The histogram of the feature values is presented in Figure 3e. The value "GPS" is found in 85.6% of the whole dataset (see Table 8) and 5.3% of the feature values are missing (see Table 10), which means that only 9.1% of the data set have non "GPS" values and therefore the feature was discarded from further processing.

Destination - 14.4% of the total number of feature values are missing. These missing values will be imputed with separate unique value and the whole feature is included in the final list for further processing.

Data source type - Typically, the value is "AIS" (see Figure 3c and Table 8). The value has no impact on the dissertation topics and the feature was removed from further processing.

Table 10: Missing value spread across the data set

Feature	Missing values	Missing values ratio	Vessels missing values	Vessels missing ratios
MMSI	3774	0.000038		
Navigational status	5960962	0.060674	461	0.11781
ROT	23411553	0.252161	986	0.25198
SOG	17983	0.000183	12	0.00307
COG	462048	0.004703	118	0.03016
Heading	2440388	0.024840	657	0.16790
IMO	29327299	0.298511	976	0.24942
Callsign	2184153	0.022232	311	0.07948
Name	1948884	0.019837	301	0.07692
Cargo type	87900086	0.894700	3625	0.92640
Width	50477	0.000514	363	0.09277
Length	48609	0.000495	359	0.09175
Draught	1047433	0.010661	689	0.17608
Destination	14183899	0.144372	718	0.18349
Vessel type	4234160	0.042769	293	0.07487
Type of position fixing device	5244781	0.053409	210	0.05676
Note. AIS data collected values treated as missed are: NaN, Unknown, Unknown value, Not under command, Not used, Undefined, No additional information, "========", "-", ".", ">", ":"				

2.4 Data Down-Sampling

A vessel's AIS transceiver sends data every 2 to 10 seconds and that depends on a vessel speed while underway, or each 3 minutes when a vessel is anchored. In practice, databases typically store data at various time intervals between subsequent registration of vessel position in the AIS system. Registration interval may vary from 2 seconds to 10 minutes and that depends on data provider. With the view to setting up the experiment, it is necessary to set the same time interval for all positions of all vessels in the same training data set. The proposed method down-samples a vessel subsequent navigational vectors to predefined interval $\Delta T_{interval}$. To achieve that, the nearest neighbour algorithm is applied to select the nearest navigational vector (Euclidean distance) according to the vessel data vector sent to AIS timestamp.

The predefined calibrated parameter $\Delta T_{interval}$ of 2 minutes was chosen. This parameter can be calibrated based on Maritime Situational Awareness (MSA) requirements of VTS. In this research it is assumed that anomaly detection will be performed in the middle range of vessel trajectory, that is on average of 20% of activity in the region of interest. On average vessels pass the investigated "Fehmarnbelt" region in 8 to 12 hours. Thus the minimum time to detect trajectory anomaly is between

1.6 and 2.4 hours. So, if $n' = 50$, then 2 minutes window falls within the sensitivity range. The nearest neighbour algorithm was applied to down-sample and obtain the feature values.

2.5 Imputation of Missing Values

There are missing feature values in the collected vessel traffic data. Depending on the type of missing feature, a different scheme for imputation has to be chosen.

- *Static features.* Features that are static and belong to the same vessel (data with the same MMSI). Such physical properties cannot change in time. For example, it can be a vessel type, length or other physical property that is sometimes distorted by the radio transmission.

- *Dynamic features.* Feature values that change in time for navigational vectors of the same vessel. It can be vessel location, heading or other data from on board vessel sensors.

- *Partially missing values of static features.* Wrong static feature values $x^{(j)}$ are available at least in few of s_v vessel navigational vectors. Examples of such features are the type of vessel, the length of the vessel or other physical parameters of the vessel. The aforementioned discrepancies happen because of the inconsistent input of data into the AIS transmission equipment.

- *Completely missing values of static features.* Feature values of the $x^{(j)}$ that are absent in the entire set of vectors s_v of a particular vessel.

- *Weakly correlated missing values of dynamic feature.* These features have a weak or very weak correlation with other features. For example, vessel rate of turn, estimated time of arrival at the port of destination, etc.

For each group of missing value types, a different value correction strategy is applied. The applied correction strategies are:

- *Partially missing static feature values.* The missing values are imputed by searching for an actual value in navigational data of the same vessel. After it is found, the rest of the vectors are imputed with that value, otherwise they are treated as completely missing static features.

- *Completely missing static features values.* The strategy depends on type and property of features and therefore there are two techniques: imputation of missing values using a predictive model, that is based on vessel trajectory and can predict missing feature in significant accuracy, that is higher than the critical threshold value T_{acc}. In this paper, the threshold of 0.95 was used. Such an approach was proposed in previous work [35]; this technique either discards the entire attribute from all data for all vessels, or, if the number of missing values is low and mostly relates to small amount of vessels, then the technique drops particular vessels from the data set.

- *Strongly correlated missing dynamic feature values.* Missing features that are strongly correlated with other features may be discarded for all vessels if they are missing in significant number of vessels. Otherwise, if only 1% of vessels lack such feature, then all navigational data of particular vessels that lack values are dropped from the data set.

- *Weakly correlated missing dynamic feature values.* If less than 1% of vessels lack particular feature, data of these vessels are excluded from further analysis. If the percentage is higher than 1% then only that specific feature is excluded from the feature list.

However, based on previous research [35], the minimum set of features must be as follows: longitude, latitude, speed over ground, course over ground, wind direction, wind speed, wave direction and height.

Imputation of missing values. Table 10 shows missing values. The imputation of missing values is performed as described in sub-chapters earlier. As mentioned in the same sub-chapter, here are two main categories of data: static and dynamic. Based on these category properties, the imputation is performed in accordance with category rules.

Static category

MMSI. These are checked vectors with high correlated features such as Name and Callsign (see Table 9). The missing MMSI values are taken from vectors with the same Name or Callsign and missing values are inserted. This way, 2571 values were imputed, other 1203 vectors were dropped as no associated Name or Callsign was found or these values were missing as well.

Width and Length. Each vessel was checked if it has at least one imputed value. This value is taken for a particular vessel and is imputed in place of missing values for the same vessel. 31457 Width and 31212 Length values were imputed in accordance with such approach. For cases where Width was available and Length was missing or otherwise, the linear regression was used to forecast the paired value. 5101 Width values and 3478 Length values were imputed using this approach. 13919 vectors had no values for either feature. These vectors were imputed with mean values of Length and Width for the same vessel type.

Draught - The missing values of Draught were imputed with help of multivariate linear regression, created with strongly correlated features Width and Length. All 1047433 missing values were imputed using this approach.

Vessel type. Vessel type feature is a special case and is described separately in sub-section 2.8

Dynamic category

SOG. This feature does not have a high degree correlation in dynamic category features, and the missing 17983 values were imputed with the mean value within the same vessel type.

COG. This feature has high degree of correlation with Heading. The missing values were imputed from Heading feature.

Voyage category

Destination. There were 14183899 missing values for the feature. Missing values were imputed with a single new category "Unknown".

Meteorological data category

All meteorological data are available without any missing values.

2.6 Data Feature Engineering

As the navigational vectors are received at different time intervals it introduces another problem. The lack of constant data intervals results in variation of sailed distance at same speed while analysing subsequent vectors of a vessel. In order to solve the issue, new differential features were introduced. The first new feature introduced is a time difference between sequential vectors' timestamps, expressed by:

$$x^{(\Delta t)}_{(v,p_v)} = x^{(timestamp)}_{(v,p_v)} - x^{(timestamp)}_{(v,p_v-1)}, \tag{3}$$

where $x^{(\Delta t)}_{(v,p_v)}$ is new feature, $x^{(timestamp)}_{(v,p_v)}$ is particular navigational data vector's time of same vessel and $x^{(timestamp)}_{(v,p_v-1)}$ is reception time of previous data vector. v is data set number of a particular vessel, p_v is vessel position vector number in sequence as in formula (2). Two more features introduced were created to express vessel movement differential in time for latitude and longitude, and are expressed by:

$$x^{(\delta Lat)}_{(v,p_v)} = \frac{x^{(Lat)}_{(v,p_v)} - x^{(Lat)}_{(v,p_v-1)}}{x^{(\Delta t)}_{(v,p_v)}} \tag{4}$$

$$x^{(\delta Lon)}_{(v,p_v)} = \frac{x^{(Lon)}_{(v,p_v)} - x^{(Lon)}_{(v,p_v-1)}}{x^{(\Delta t)}_{(v,p_v)}}, \tag{5}$$

where $x^{(\delta Lat)}_{(v,p_v)}$ and $x^{(\delta Lon)}_{(v,p_v)}$ are newly constructed features based on latitude and longitude differences in subsequent vectors $x^{(Lon)}_{(v,p_v)}$, $x^{(Lon)}_{(v,p_v-1)}$ of the same vessel.

In addition to that, earlier works have shown that meteorological data has significant influence on marine traffic models [35]. The data include information about wind direction, wind strength, swell direction, swell height, swell period, day/night, and tide level. These aforementioned features are also artificially added to each vessel data vector that was registered by AIS system. Meteorological data were taken periodically from the European Centre for Medium-Range Weather Forecasts (ECMWF) grid. ECMWF provides data in certain geographical interpolated resolution. Figure 4 represents available meteorological data locations in red dots. Vessel position accuracy is much higher than meteorological data grid, thus the assignment of a particular grid point to vessel navigational vector is accomplished. Meteorological data is assigned to a navigational vectors by using the algorithm of the nearest neighbour by location and time. At first, distance to all meteorological locations is calculated by haversine formula [58] using WGS84 geodetic system coordinates of vessel and meteorological locations. The closest meteorological location is assigned based on calculated distances and then the closest in time forecast is picked from all forecasts for this location (Figure 4). This meteorological data is assigned to vessel position vector. This way it becomes possible to assign meteorological conditions at each vessel track point in whole vessel trajectory.

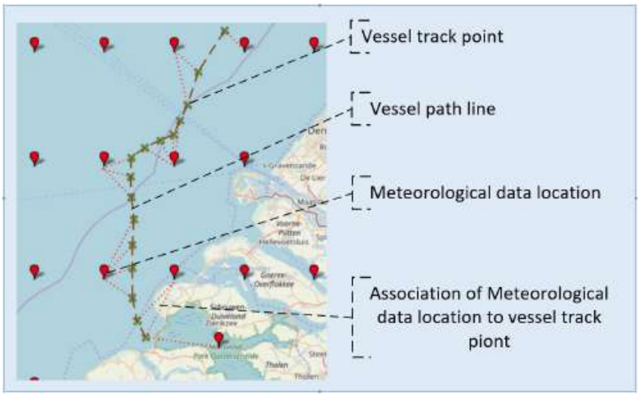

Vessel track point
Vessel path line
Meteorological data location
Association of Meteorological data location to vessel track piont

Figure 4: Meteorological data grid [57]

Final list of features The final list of features can be found in Table 11. The total number of features are $f = 15$. The features MMSI and

Table 11: Final list of features

Feature category	**Features**
Static category	Vessel size (With, Length, Draught processed with MDS)
Dynamic category	Latitude, longitude, SOG, COG
Voyage category	Destination
Meteorological category	winddirDegree, windspeedMeterSec, swellheight, swelldirection, swellperiod, day/night
Engineered category	$x^{(\Delta t)}, x^{(\delta Lat)}, x^{(\delta Lon)}$

Timestamp are used only in restructuring data set and then splitting to sequences. Feature "Ship Type" is used to split data sets to separate data sets by vessel type in order to train separate models.

2.7 Splitting Vessels Navigational Vectors to Sequences

In the following chapter for vessel position prediction, the algorithm of artificial deep neural network (DNN) is described. In order to obtain the predictions the data must be in certain three dimensional format, thus a sliding window approach for data slicing is applied. Slicing algorithm takes data set restructured matrix S function (2). Each vessel data set s_v in a window is processed separately and $\{x_{(v,1)}, x_{(v,2)}, \ldots, x_{(v,p_v)}\}$ set is

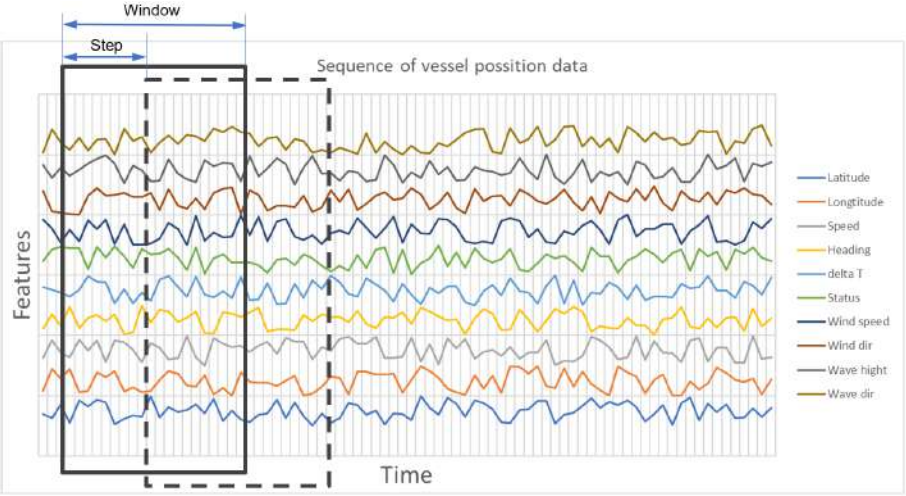


Figure 5: Sliding window visualization

sliced into sequences by length of $\widetilde{n}+n'$ (fig. 5), where $\widetilde{n}$ is length of DNN output sequence (prediction) and n' is length of input (for prediction) sequence. The window slides according to a predefined step of η each time producing a new sequence. Each sequence is divided into two parts. The first part in length of n' is assigned to input matrix, and the second part in length of $\widetilde{n}$ is written into output matrix. The obtained matrices can be defined by expressions:

$$(\chi, Y) = \left(\begin{bmatrix} x_{(1,1)} & x_{(1,2)} & \cdots & x_{(1,i)} & \cdots & x_{(1,n')} \\ x_{(2,1)} & x_{(2,2)} & & x_{(2,i)} & \cdots & x_{(2,n')} \\ & \vdots & \ddots & & \vdots & \\ x_{(g,1)} & x_{(g,2)} & \cdots & x_{(g,i)} & \cdots & x_{(g,n')} \\ & \vdots & \ddots & & \vdots & \\ x_{(N,1)} & x_{(N,2)} & \cdots & x_{(N,i)} & \cdots & x_{(N,n')} \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} x_{(1,n'+1)} & x_{(1,n'+2)} & \cdots & x_{(1,n'+r)} & \cdots & x_{(1,n'+\widetilde{n})} \\ x_{(2,n'+1)} & x_{(1,n'+2)} & & x_{(2,n'+r)} & \cdots & x_{(2,n'+\widetilde{n})} \\ \vdots & & \ddots & & \vdots & \\ x_{(g,n'+1)} & x_{(g,n'+2)} & \cdots & x_{(g,n'+r)} & \cdots & x_{(g,n'+\widetilde{n})} \\ \vdots & & \ddots & & \vdots & \\ x_{(N,n'+1)} & x_{(N,n'+2)} & \cdots & x_{(N,n'+r)} & \cdots & x_{(N,n'+\widetilde{n})} \end{bmatrix} \right) \tag{6}$$

$$g \in \{1, 2 \ldots, N\}, \quad i \in \{1, 2 \ldots, n'\}, \quad r \in \{1, 2 \ldots, \widetilde{n}\}$$

where χ is input of the model, Y is output of the model, N is number of vessel navigational vectors sequences, n' - length of single sequence for input, $\widetilde{n}$ - length of navigational vector sequence for output. , χ and Y are matrices of vessels' navigational vector sequences formed by sliding window process while assigning a vector $x_{(v,p_v)}$ from current window position to a sequence matrices (χ, Y).

Obtained matrices further are split into subsets that are used for model training, validation and testing. Data split is organized by random rows selection.

This method was applied on each vesse's sequential (time series) vectors structured by S function (2). The splitting window sizes were set to $n = 50'$ and $\widetilde{n} = 50$. The quantity of prepared sequences by vessel type (feature "Vessel type") can be found in the Table 12 under unbalanced sequences columns.

2.8 Classification of Vessel Types

The lack of the data such as vessel type prevents the creation of a sufficiently accurate model for detection of unusual vessel traffic. It is therefore necessary to develop a method for imputation of the missing data.

Table 12: Split sequences of vessels before and after class balancing

	Sequences, unbalanced classes		Sequences, balanced class (SMOTE)			
Vessel type	Vessels	Total	Total	Train	Validation	Test
Cargo	1763	110554	110554	66332	22111	22111
Tanker	585	33005	110554	66332	22111	22111
Fishing	75	23022	110554	66332	22111	22111
Passenger	73	70153	110554	66332	22111	22111
Tug	641	3773	110554	66332	22111	22111
Military	54	6934	110554	66332	22111	22111
Sailing	52	9783	110554	66332	22111	22111
Dredging	38	6701	110554	66332	22111	22111
Pleasure	32	2873	110554	66332	22111	22111
SAR	29	5415	110554	66332	22111	22111
Pilot	21	9287	110554	66332	22111	22111
Towing	13	764	110554	66332	22111	22111
Reserved	13	639	110554	66332	22111	22111
Law_enforcement	12	6530	110554	66332	22111	22111
Towing_long_wide	11	538	110554	66332	22111	22111
HSC	8	47	110554	66332	22111	22111
Port_tender	5	398	110554	66332	22111	22111
Diving	5	152	110554	66332	22111	22111
Anti-pollution	2	1600	110554	66332	22111	22111
Spare_1	1	23	110554	66332	22111	22111
WIG	1	59	110554	66332	22111	22111
TOTAL	2857	302250	2321634	1392972	464331	464331

In this subsection a technique for imputation of missing vessel type data is described. The imputation helps to improve detection of abnormal maritime traffic. The results of this research were presented in paper [35] by Venskus and Treigys.

Figure 3a shows histogram of vessel types. All missing vessel types are marked as "undefined" and assigned to a single group. We see that the number of missing vessel type values is 4234160 and that constitutes 4.28% (see Table 10) of all data in this region during the specific time period. Figure 6 shows marine traffic visualisation in geographical WGS84 plane as a two dimensional projection. Different colors represent traffic of different vessel types. It can be observed that vessel types have distinctive patterns. Based on that, the model of imputation of the missing "Vessel type" values is created.

In order to recognize a vessel type from available navigational vessel vectors, a model, which was trained on available data, needs to be developed and later this model should be used to classify types of unknown vessels. Figure 6 presents different patterns of marine traffic of specific vessel types (visualized vessel type traffic can be found in Appendix B).

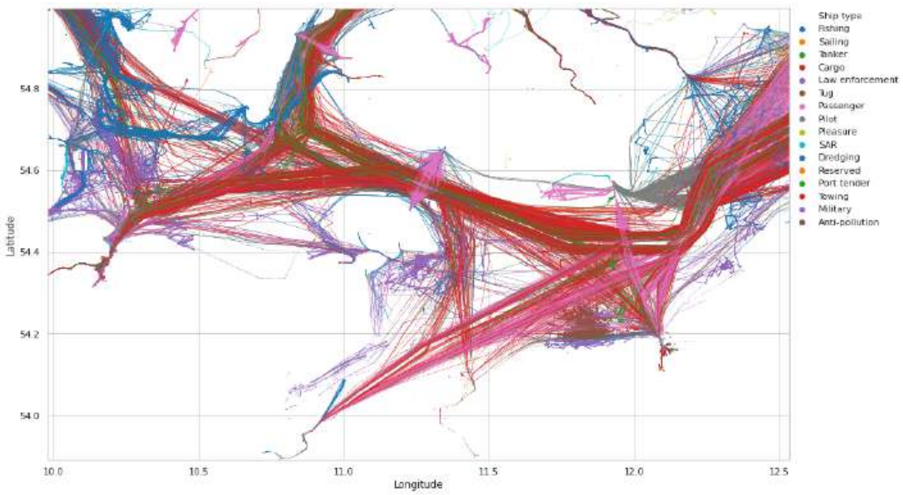


Figure 6: Traffic of different vessels types in geographical coordinate plane

For the classification task, the data must be prepared as described in earlier subsections. The prepared data is split in two groups: the data with known and unknown vessel types. The data with known vessel type data will be used for model training, validation and testing. Vessel type in data set is assigned to each vessel's navigational vectors sequence as a class to be learned. The model will be trained to recognise vessel type as a class based on vessel's navigational vectors sequence, i. e. vessel trajectory.

In order to teach a model to classify vessel types, a deep neural network is constructed (see Figure 7). The input of neural network is a sequence of single vessel's navigational vectors, that have been constructed as described in previous subsections. The neural network input layer is of multi-step and multivariate type, which is represented as two dimensional matrix. The first dimension contains vessel's navigational vectors at sequential time steps. The second dimension contains multivariate features for time steps such as longitude, latitude, Speed Over ground (SOG), Course Over ground (COG), wind direction, wind speed, wave height, etc. The detailed list of used features is described in Table 11(34 p.) in subsection 2.6. The dimensions of matrix is $n' \times f$, where n' is length of a sequence (time steps) and f is a number of features. The deep neural network architecture has two main modules, namely, a multi-stacked multivariate LSTM network and multi-layer perception models. The input layer is interconnected sequentially with first layer of LSTM module. Each LSTM layer has Q_{lstm} number of LSTM cells in each out of f number of feature sublayers. The number Q_{lstm} of LSTM cells is

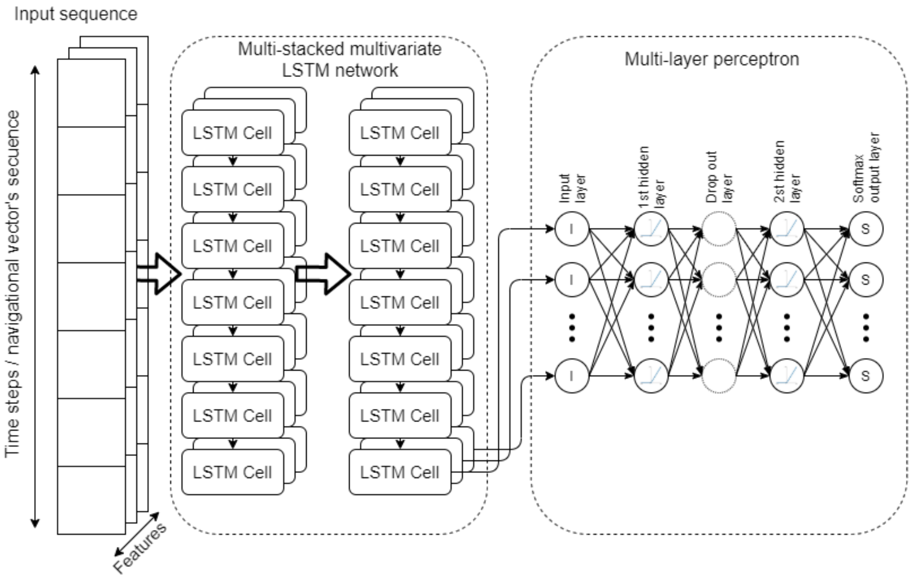


Figure 7: Scheme of vessel type classifier

determined during tuning of the neural network in order to gain maximal accuracy. The description of LSTM cell can be found in subsection 4.1 under paragraph *Long short term memory neural network*. The LSTM layers are stacked on top of each other and are interconnected sequentially. The bottom layer cell output $y(t)$ is provided as a top cell input $x(t)$. The last LSTM layer cell output is connected to multi-layer perceptron module through it's input layer. The input layer has f number of neurons. The Multi Layer Perceptron (MLP) has two hidden layers with Q_{mlp} neurons each. These layers are interconnected through drop out layer for network overfitting regularization [59]. Hidden layer uses Rectifier Linear Unit (ReLU) as activation function. The MLP output is softmax layer [60], where each neuron represents class probability.

The workflow of vessel type classification can be summarised as:

1. The raw data is prepared as described in subsection 2 *Data Preparation*.

2. A vessel type is assigned as a class feature to each vessel's navigational vectors sequence. The missing vessel type values are marked as "undefined" and are separated from data. Then the data set with vessel types is evaluated for class imbalance. As is seen in vessel type histogram (Figure 3a), the vessel type classes are strongly imbalanced. For imbalance handling a Synthetic Minority Oversampling Technique (SMOTE) technique is used [61].

3. Before network training, the data sets are randomly shuffled and separated into three subsets. 50% of data set are used for neural network (Figure 7) training. 30% of data are used for validation and model fine tuning during a neural network training and the third set with 20% of data is used for model testing and general error estimation.

4. Initial $Q_{lstm}^{(min)}$ number of LSTM cells in a layer is chosen, and the network is trained. As a loss function, the Sparse Categorical Cross Entropy [62] with the stochastic optimizer Adam [63] were used.

5. Model weights and prediction values are stored for further process. The accuracy of the trained network is evaluated using validation data set. If Q_{lstm} hasn't reached maximum value $Q_{lstm}^{(max)}$, then the workflow is repeated from step 4.

6. Finally, the most accurate model is chosen to predict a vessel type that matches certain vessel's navigational vectors sequences.

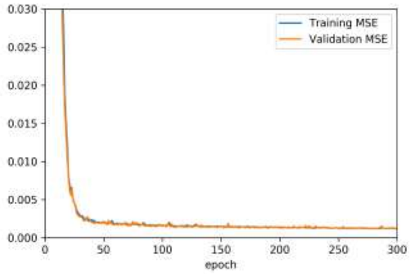


Figure 8: Loss trend of vessel type classifier training

The model parameters are set as follows: the LSTM cells per layer is set to $Q_{lstm} = 250$; Neurons per MLP hidden layer is set to $Q_{mpl} = 40$; the dropout layer's rate is set to 0.2. The networks loss function categorical cross entropy is used. The Adam optimiser [63] was used with the following parameters: $learning\_rate = 0.001, beta\_1 = 0.9, beta\_2 = 0.999, epsilon = 10^{-07}$. The Figure 8 depicts training and validation losses over epochs. A rapid loss drop is observed until 45th epoch and then a slow dropping is seen until 250 epoch, and after that it becomes stable. Tables 35 and 36 in Appendix C show confusion matrix of a trained vessel type classifier.

Table 13 shows evaluation measures of vessel type recognition model. The evaluation was performed using test data set, which was kept separate during the whole training process. The five classification measures

Table 13: Evaluation measures of vessel type recognition model

Vessel Type	Precision	Sensitivity	F1-score
Cargo	0.96727	0.97182	0.96954
Tanker	0.97004	0.98268	0.97632
Fishing	0.95735	0.96739	0.96234
Passenger	0.96518	0.96034	0.96275
Tug	0.85723	0.87490	0.86597
Military	0.96387	0.97838	0.97107
Sailing	0.95072	0.94405	0.94738
Dredging	0.96486	0.97852	0.97164
Pleasure	0.96388	0.96201	0.96295
SAR	0.99268	0.99299	0.99283
Pilot	0.91491	0.90887	0.91188
Towing	0.86471	0.88716	0.87579
Reserved	0.99896	0.99928	0.99912
Law_enforcement	0.97808	0.94057	0.95896
Towing_long_wide	0.95590	0.87545	0.91391
HSC	0.98834	0.99299	0.99066
Port_tender	0.94639	0.95798	0.95215
Diving	0.97736	0.99977	0.98844
Anti-pollution	0.99900	0.99860	0.99880
Spare_1	0.99991	0.99910	0.99950
WIG	0.99995	0.99973	0.99984
Average	0.96079	0.96060	0.96056
Accuracy			0.96060

were calculated: precision, sensitivity, f1-score for each class, average, and accuracy for all classes. It is observed, that classification of vessel types Tug, Towing, and Towing_long_wide shows lower precision values than other classes, i. e. 0.85723, 0.86471, and 0.95590 per vessel class accordingly. Moreover, by careful analysis of fusion matrix (Appendix C) one can state that these three class representatives share false positives and negatives mostly between themselves, which means that distinction of the vessel from supplied category is ambiguous. The class Tug was falsely negatively classified as Towing 1089 times, similar behaviour observed with Towing_long and Tug (1199 false negative cases) vessels. Other groups of false negative classes include Military vs. Law enforcement and Sailing vs. Pleasure vessel types. Short analysis shows that these classes have the most similar traffic behavior in maritime domain.

The precision average, sensitivity, f1-score (see Table 13) of vessel type classifier are high enough to impute missing vessel types.

Table 10 depicts 4234160 missing vessel type vectors, which constitute 4.3% of all vectors of the "Fehmarnbelt" data set. Table 14 summarizes the final list of data sets grouped by vessel type. Finally, each

Table 14: Prepared final data sets of marine vessel traffic

Vessel type	Vessels count	Sequences				Total vectors
		Train	Validation	Test	Total	
Cargo	1944	75625	25208	25209	126042	12604200
Tanker	645	22577	7526	7526	37629	3762900
Fishing	83	15748	5249	5250	26247	2624700
Passenger	80	47988	15996	15996	79980	7998000
Tug	71	9421	3140	3141	15702	1570200
Military	59	4743	1581	1581	7905	790500
Sailing	57	6692	2231	2231	11154	1115400
Dredging	42	4584	1528	1528	7640	764000
Pleasure	35	1965	655	655	3275	327500
SAR	32	3704	1235	1235	6174	617400
Pilot	23	6352	2118	2118	10588	1058800
Towing	14	522	174	175	871	87100
Reserved	14	436	146	146	728	72800
Law_enforcement	13	4467	1489	1489	7445	744500
Towing_long_wide	12	367	123	123	613	61300
HSC	9	32	11	11	54	5400
Port_tender	6	272	91	91	454	45400
Diving	5	103	35	35	173	17300
Anti-pollution	2	1094	365	365	1824	182400
Spare_1	1	15	5	6	26	2600
WIG	1	40	13	14	67	6700
TOTAL	3148	206749	68917	68925	344591	34459100

vessel type data set is divided to train, validate, and test subsets with rations 80:20:20, respectively. The test set is kept untouched in anomaly detection model creation and is used only for final model evaluation. The same data sets are used for investigation in all methods, algorithm investigation and evaluation.

2.9 Conclusions of the Section

In this section, AIS and meteorological data were related to Klaipeda and "Fehmarnbelt" sea regions. Unprocessed data from the "Fehmarnbelt" data set consisted of 98245370 records, Klaipeda sea area had 642541, and meteorological data aggregated 20608 records.

The multistacked multivariate LSTM classifier was developed to cope with the issue of missing vessel type. The proposed model performs very well. After inspection of per class precision (in most cases higher than 0.96), recall, f1-score metrics show good generalization properties allowing to gain classes for lacking 4.28 percent (4234160 navigational vectors) of the data in the "Fehmarnbelt" data set.

Data preparation was performed for upcoming anomaly detection analysis. Preprocessing includes data structuring, cleaning, down-sampling, imputation of missing values, feature engineering, and splitting to sequences of vessel navigational vectors. According to the different nature of the AIS data, an appropriate data imputation scheme has been introduced. Overall, the applied data prepossessing resulted in total "Fehmarnbelt" data set of 34459100 records, Klaipeda's sea area data set with 232093 records.

Finally, the research assumes that anomaly detection will be inspected in the middle range of vessel trajectory, which is an average of 20% of activity in the region of interest. On average, vessels pass the investigated "Fehmarnbelt" region in 8-12 hours. Thus, the minimum time to detect trajectory anomalies is between 1.6 and 2.4 hours. To this end, calibrated $n' = 50$ allows to achieve vessel trajectory inspection sensitivity of 2 minutes.

3 Semi-supervised Point Based Vessel Traffic Anomaly Detection

This section presents the description of point based semisupervised marine vessel traffic anomaly detection based on SOM and virtual pheromone integration [26], and SOM with Gaussian Mixture Model (GMM) [56]. The section describes design details of these methods, selection of parameters and retraining strategies. These researches were presented in papers by Venskus *et al.* [26, 27].

3.1 Maritime Anomaly Detection Using an Integration of a Self-Organizing Map and a Virtual Pheromone

3.1.1 Clustering with SOM

The self-organizing map (SOM) is a neural network-based method that is trained in an unsupervised way using a competitive learning [64, 55]. A distinctive characteristic of this type of neural networks is that they can be used for both visualization and clustering of multidimensional data. The most important property of SOM can be utilised for many tasks, such as reduction of the amount of data, speeding up learning nonlinear interpolation and extrapolation, generalisation, and efficient compression of information [65]. SOM is one of the most analyzed neural networks, that is learned in an unsupervised manner. In our case, SOM represents a set of neurons, connected to one another via a rectangular topology. The rectangular SOM is a two-dimensional array of neurons $W = w_{ij}, i = 1, \ldots, k, j = 1, \ldots, s$. Here k is the number of rows, and s is the number of columns. Each element of the input observation vector is connected to every individual neuron in the rectangular structure. Any neuron is entirely defined by its location on the grid by its specific index at the row i and the column j, and by its weight (so-called code book) vector. After SOM training, the data are presented to SOM and the winning neuron for each data vector is found. The winning neuron is the one to which the Euclidean distance of the input data vector is the shortest. This way the data vectors are distributed on SOM, and some data clusters can be observed.

The results of a SOM map depend on the selected learning parameters. Learning rates and neighbourhood functions h_{ij} are the necessary parameters that influence the results. The neighbourhood function determines how strongly the neurons are connected to each other and influences the training result of SOM. Therefore, it is important to choose the proper neighbourhood function. There are different kinds of neighbourhood functions: bubble, Gaussian, Cut Gaussian [66, 67], heuristic

Table 15: Neighbourhood functions

Gaussian	$h_{ij}(t) = \exp\left(-\frac{d_{ij}^2}{2(\eta_{ij}(t))^2}\right)$
Bubble	$h_{ij}(t) = F\left(\eta_{ij}(t) - d_{ij}\right)$
Cut Gaussian	$h_{ij}(t) = \exp\left(-\frac{d_{ij}^2}{2(\eta_{ij}(t))^2}\right) F\left(\eta_{ij}(t) - d_{ij}\right)$
Triangular	$h_{ij}(t) = \begin{cases} 1 - \frac{\lvert d_{ij}\rvert}{\eta_{ij}(t)}, \text{ if } \lvert d_{ij}\rvert \leq \eta_{ij}(t) \\ 0, \text{ otherwise} \end{cases}$
Mexican hat	$h_{ij}(t) = \left(1 - \frac{d_{ij}^2}{(\eta_{ij}(t))^2}\right) \exp\left(-\frac{d_{ij}^2}{2(\eta_{ij}(t))^2}\right)$

[68], Mexican hat [69], triangular [69], rectangular [69] and others. In this research, I have compared five neighbourhood functions and their influences on the classification results obtained by the modified SOM method. These functions are presented in Table 15, where d_{ij} is a distance between the current observation vector and the winning neuron, η_{ij} is the neighbourhood radius, F is a step function: $F(x) = 0$, if $x < 0$ and $F(x) = 1$, if $x \geq 0$.

As mentioned before, the learning rate also influences the results of SOM. Usually, linear, inverse-of-time, and power series learning rates are used for the SOM training [70, 71]. In this research, the learning rate is constant and equal to 0.5, both the initial neighbourhood radius and the radius decay parameters are set to -0.1.

3.1.2 Classification by Using a Virtual Pheromone Concept

The application areas of SOM are data clustering and graphical result presentation. In this subsection, the Thesis proposes to exploit the biologically-inspired notion of a virtual pheromone to use the collected knowledge about clusters and classify marine traffic abnormality. The idea is based on the observations of ant colonies. To mark the way to the food source, the ants use a chemical substance called pheromone. Other ants follow the pheromone trail to reach the discovered food source. Pheromone evaporates in time, and the trail on the road slowly disappears. The ants must continually travel by the same route to strengthen the evaporating pheromone trail.

In the proposed approach, a modified SOM method's training process is the same as the original one except that the virtual pheromone intensity value is introduced in the last epoch. When the SOM network training is completed, all possible training and validation data vectors are shown to SOM. Further, considering the number of vectors assigned to the same cluster, it can be calculated how this cluster represents a majority. It is necessary to count the number of vectors in the cluster in order to assign the vectors from the training data set to winning neurons. The number of represented data vectors by SOM neuron is written next to the weight of the winner neuron (codebook). This value is called virtual pheromone mark Q.

In the beginning, each SOM neuron has its pheromone intensity value, which is equal to the cluster size (number of vectors in that cluster) of this neuron. The value of a pheromone mark Q is calculated as follows: when the winning neuron is selected for the individual data vector, the pheromone intensity value is increased by one, i. e., the pheromone mark is associated with the appropriate neuron. Thus, the more data set vectors are assigned to the same winning neuron, the higher its virtual pheromone intensity is.

In order to adjust the pheromone evaporation procedure, after each SOM network re-training, the virtual pheromone intensity value τ_{ij} is updated according to the equation:

$$\tau_{ij}(t_2) = (1-\rho)\cdot\tau_{ij}(t_1) + Q_{ij}, \tag{7}$$

where τ_{ij} is a virtual pheromone intensity; t_1 is the previous state of the virtual pheromone intensity; t_2 is the recent state of the virtual pheromone intensity. The parameter ρ represents a virtual pheromone intensity evaporation speed ($0 < \rho < 1$). In the formula, similarly to the ant colony system, the pheromone trail will evaporate unless it is renewed within a particular time. The intensity reduction is slower than its renewal process [72], [73].

The pheromone intensity threshold used for abnormal movement detection is calculated using the validation data set. The precision and sensitivity of the algorithm can be adjusted by changing the threshold value. To adjust the threshold, the classification error cost function has been optimized according to:

$$J(\Theta) = -\beta_{PPV}\log(PPV_\Theta) - \beta_{TPR}\log(TPR_\Theta), \tag{8}$$

$$PPV_\Theta = \frac{TP_\Theta}{TP_\Theta + FP_\Theta}, \tag{9}$$

$$TPR_\Theta = \frac{TP_\Theta}{TP_\Theta + FN_\Theta}. \tag{10}$$

Here $J(\Theta)$ is a classification error rate and Θ is a threshold value; PPV_Θ is classification precision; TPR_Θ is classification sensitivity; β_{PPV} and β_{TPR} are the influences of classification parameters on the classification error cost function; TP_Θ is the count of true positive (assigned to an abnormal state) observations; FP_Θ is a false positive (assigned to a false abnormal state); TN_Θ is a true negative (assigned to a normal state); FN_Θ is a false negative (assigned to a false normal state). The gradient descent method has been used to find a local minimum of the function expressed by equation (8).

When the new vessel position is received, the navigational vector representing this state is assigned to the appropriate SOM cluster by choosing the winning (closest) neuron of this cluster. Thus, a winning neuron and its representing cluster should be found for each newly received vessel navigational vector. This vessel vector is classified based on the pheromone value assigned to SOM neuron representing the assigned cluster. The vessel vector is classified as normal if the pheromone value is greater than the threshold value Θ, or abnormal if less.

3.1.3 Method Description

As mentioned earlier, the combination of a self-organizing map and a virtual pheromone is proposed to classify events for abnormal movement detection in maritime traffic. It is important to identify whether the observation data show the abnormal vessel behaviour and to react accordingly. Therefore, creating and testing the algorithm, the trained SOM neural network is transferred to a system where it classifies real time data, based on the existing network settings without additional retraining. However, as the amount of new data increases, in order to ensure a high classification accuracy, there is a necessity to re-train the network periodically. The re-training process of the neural network is run by adding new observation data to the training set.

A general scheme of the proposed algorithm (SOM_Pheromone) is presented in Fig. 9. Its implementation steps are described as follows:

- *Data processing.* The data filtering is applied in order to reject repeated and erroneous data, then the data set is divided into three subsets: training, validation, and testing.
- *Normalisation of the training data set.* Each observation attribute is scaled with Min-Max to interval from 0 to 1.
- *SOM network training.* Each winning neuron has its pheromone intensity value which is equal to the number of data vectors assigned to the winning neuron. The virtual pheromone value is calculated in the last epoch. During the SOM re-training process, the function

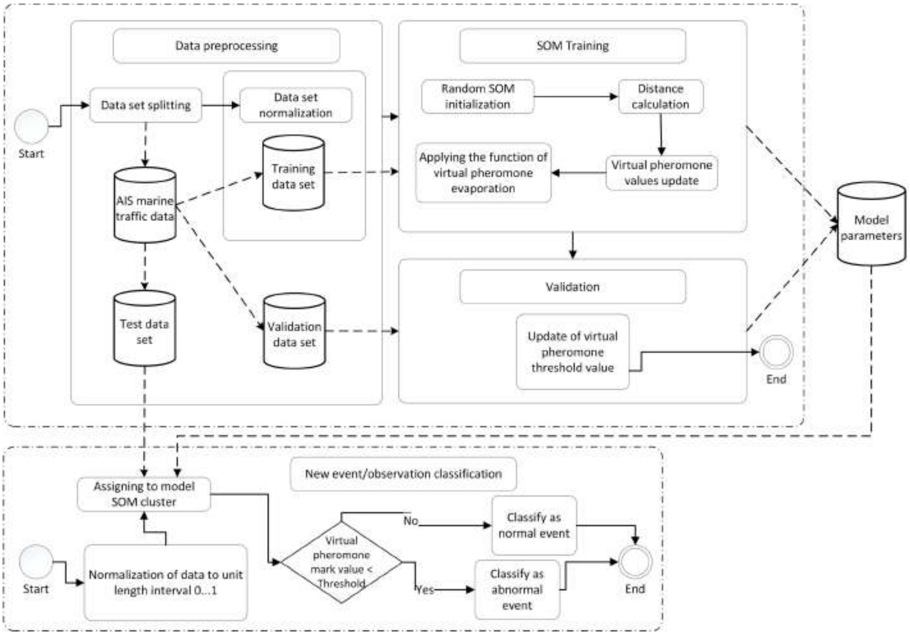

Data preprocessing
Start
Data set splitting
Data set normalization
AIS marine traffic data
Training data set
Test data set
Validation data set
SOM Training
Random SOM initialization
Distance calculation
Applying the function of virtual pheromone evaporation
Virtual pheromone values update
Model parameters
Validation
Update of virtual pheromone threshold value
End
New event/observation classification
Assigning to model SOM cluster
Start
Normalization of data to unit length interval 0...1
Virtual pheromone mark value < Threshold
No
Yes
Classify as normal event
Classify as abnormal event
End

Figure 9: Integration of a self-organizing map and a virtual pheromone

of the virtual pheromone intensity evaporation is applied.

- *Tuning of the pheromone threshold using validation data.* The sensitivity and precision of the algorithm are adjusted by changing the threshold value. After the SOM network training, the threshold value of the pheromone intensity for abnormality detection is chosen with respect to the minimum and maximum values of pheromones. To adjust the optimal threshold, the classification error rate function has been used (see eq. 8).
- *Testing of the algorithm using test data.* The test data set is normalised to interval 0 and 1. Further, the test data observations are classified as normal or abnormal by taking into account the SOM network parameters and pheromone values obtained in the training step.
- *Classification of new marine traffic observations.* The classification of new data in real time is based on the resulting network settings without additional SOM training.

3.2 Maritime Anomaly Detection Using Self-Organizing Maps and Gaussian Mixture Models

Comparisons of the proposed algorithm with other similar methods are performed. In this subsection the additional SOM-based algorithm for abnormal movement detection in marine traffic is presented [14], [74]. This anomaly detection method (SOM_GMM) is a combination of SOM and Gaussian Mixture Models (GMM). Figure 10 depicts the implementation steps of the algorithm. Those steps are described as follows:

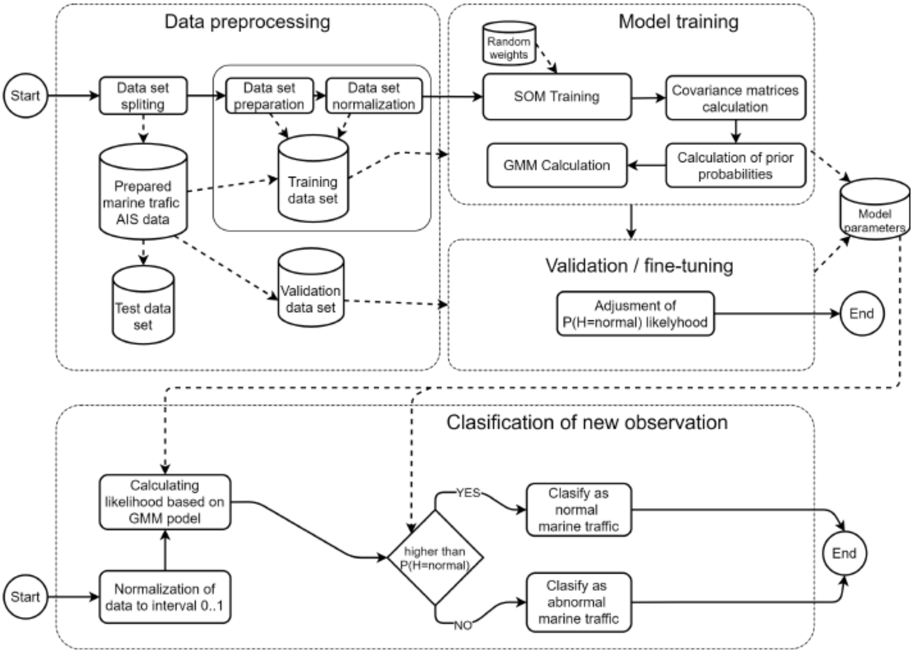

Data preprocessing
Model training
Start
Data set spliting
Data set preparation
Data set normalization
Random weights
SOM Training
Covariance matrices calculation
Calculation of prior probabilities
GMM Calculation
Prepared marine trafic AIS data
Training data set
Model parameters
Test data set
Validation data set
Validation / fine-tuning
Adjusment of P(H=normal) likelyhood
End
Clasification of new observation
Calculating likelihood based on GMM podel
YES
Clasify as normal marine traffic
higher than P(H=normal)
End
Start
Normalization of data to interval 0..1
NO
Clasify as abnormal marine traffic

Figure 10: Integration of a SOM and a Gaussian Mixture Model

- *Division of the available data sets.* The available vessel traffic data from the area of interest are divided into three sets: 50% - a training data set for SOM learning, 30% - a validation/adjustment data set for a pheromone intensity threshold calculation, and 20% - a test data set for evaluation of classification results.
- *Pre-processing of the training data set.* During the pre-processing, all the duplicate data vectors are filtered out.
- *Normalisation of the training data set.* Each attribute has been normalised into the range of 0 and 1.
- *SOM calculation.* The learning process of SOM is influenced by several parameters: shape of the grid is square, the learning rate was set to 0.5, the weight range was set to 0.5. The Gaussian neighbourhood function was used. Both the initial neighboring radius and the radius decay parameters were set to -0.1.
- *Covariance matrix calculation.* For each map neuron, the covariance matrix of all input vectors that correspond to a winning neu-

ron is calculated.

- *Calculation of prior probabilities.* For each SOM cluster the n-dimensional Gaussian probability density function has been calculated. The mean of each density function corresponds to the weights of the SOM neuron vector, and the variance is given by dispersion of training data.
- *GMM calculation.* GMM is calculated by summing all Gaussian distributions of each SOM cluster.
- Adjustment of the P(H=normal) likelihood value on validation data set.

In paper [14], a division of the anomaly detection process into on-line and off-line sub-processes has been proposed. The on-line data processing refers to the analysis of incoming data in real-time, the off-line processing relates to the establishment of normal models from the training data and rules used during the on-line detection process. The method, presented in papers [14], [56], is based on two assumptions: unusual events have to be sufficiently different from the normal events in order to be detectable; the training set should be free from unusual events. The same assumptions and conditions were met while carrying out the experiments described in Section 5.

3.3 SOM retraining strategies

This subsection presents SOM retraining strategies for marine traffic anomaly detection. It is a part of doctoral research presented by Venskus *et al.* in paper [27].

Motivation in SOM retraining strategies. With the growth of maritime traffic, especially near seaports, the complete retraining of the SOM algorithm becomes costly in terms of training time. The need for algorithm retraining is quite straightforward: the more vessel movement data are observed and fed into the algorithm, the better the precision of the algorithm should be. All neural networks are strongly dependent on the input sequence in the training data. It was observed that, if only the input sequence of the data changes, even though the system architecture stays the same, classification accuracy results may be significantly impaired [75]. Other authors proposed neural networks retraining strategies to build compact neural network models with less memory usage and faster inference speed [76]. Recently, the SOM neural network is being used to build data sets used in deep neural network model retraining [77, 78] or is used as a part of deep neural network model [79]. Different areas of applications of the SOM algorithm depicts the necessity to investigate algorithm effectiveness more thoroughly with respect to algorithm sensitivity, precision and data processing time by introducing different retraining strategies. SOM retraining ensures the inclusion of the most recent movement data that reflects actual conditions and context. To maintain high algorithm precision and sensitivity, approaches to data streaming, batching and model retrain strategies have to be explored [80].

Introduction of retraining strategies. In this dissertation, two neural network retraining strategies are presented. The research and comparison of the results with the standard procedure of neural network model experimental investigation (so-called Strategy I) is presented by Venskus *et al.* in paper [27].

- Strategy I presents data batching and algorithm training whenever the new batch becomes available as if no model history data were available. It is a common approach for neural network training/validation/testing. It is used as a reference in order to compare retraining Strategies II and III introduced by Venskus et al. in paper [27].
- Strategy II presents algorithm performance while using pre-trained model parameters on previously trained data with the new incoming data batches.
- Strategy III presents different data batch shuffling techniques and the use of previously pre-trained model parameters.

All three strategies were investigated for the learning rate parameter influence on the model performance and training time as well. Data passed from a vessel can be viewed as a stream that contains facts regarding vessel movement trajectories. Those may depend on seasonal data, the shipping routes, schedules, and so on. Thus, the abnormality detection model has to be developed by analyzing vessel movement trajectories (as well as historical data) in an incremental manner based on the up-to-date data.

Preparing batches. First, 20% of the vessel type data set is randomly selected for the general model error evaluation. Then, the resulting 80% of the data set items are used for the data batching strategy. These 80% of data items were split into 20% for strategy testing, and 80% for T1, T2, and T3 data batch splitting (see Figure 11) to perform the SOM network training and validation. Batches were used in the experiments to imitate the continuous data arrival with the view to investigate different SOM

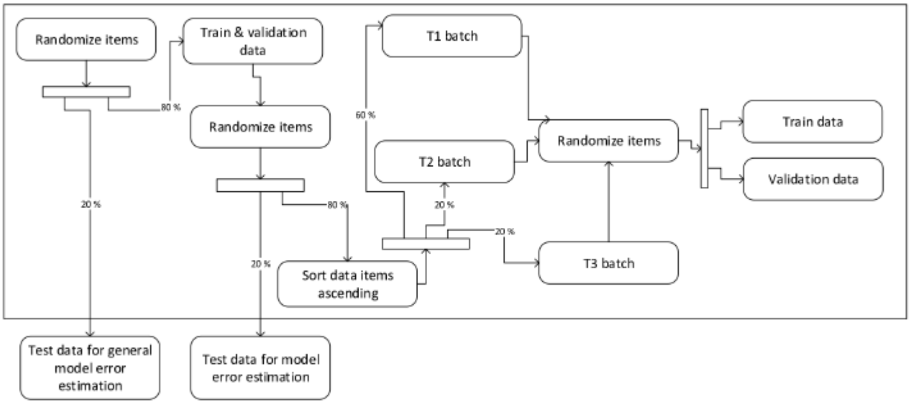


Figure 11: Data split scheme [27]

network retraining strategies and learning rate parameter selection. The scheme of data split is shown in Figure 11.

All data items were sorted in ascending order with respect to data sending timestamp. The SOM network of size 60×60 was taken according to the SOM size method investigation published in [26].

Strategy I. In Strategy I for the SOM network training and validation, T1, T2 and T3 data batches are used. The learning rate parameter is set to 0.5. Then, after the network has been trained and validated with the T1 data batch, the new data were fed to the network as follows: the T1 and T2 batch data were merged together and the algorithm was trained from the initial random state using all items from T1 and T2. The same scheme was applied to the T3 data batch. To get the best network performance, the learning rate parameter can be adjusted. This way, the training experiment of Strategy I is repeated while every learning parameter value is tested to achieve the best algorithm performance.

After the model is trained, it is tested with the test data set, which allows to evaluate the general model error. The best obtained model characteristics are chosen using the test data set.

Strategy II. The initial algorithm of strategy I is trained 10 times with the T1 batch data. During each training, the weights of the SOM network were generated randomly, and the best performing network was selected while keeping a fixed learning rate parameter at the value of 0.5. Then, the best obtained network parameters were used as initial weights for the network to be trained with T2 batch data.

Finally, imitating the new data portion arrival, the best model obtained with T2 batch data was retrained with the T3 batch data. The results of research [27] show that the best performance network was obtained with learning rate 0.025.

Strategy III. The scheme of the model training validation and testing was similar to that described in Strategy II, except for the following two aspects. Firstly, from T2 and T3 batches, four data batches (Tm2–Tm5) were produced, and each batch contained one quarter of both T2 and T3 data. Secondly, as previously described, after every model training and validation, the parameters of the best obtained model were used for every next Tm2–Tm5 batch training, except for the model training data aggregation. For every retraining a test data for model error estimation of data was used as described in previous Strategies I and II. Half the items from Tm2–Tm5 data batches consisted from items from T2 and T3, (Tm2–Tm5) while another part of the data was selected proportionally, with respect to those data points attached to the previous best model SOM winning neurons. This approach guaranteed that the knowledge of frequently passed sea regions was incorporated into the next model training because it is not frequent for the vessels to change their sea routes.

The experimental study of the strategies on "Klaipeda" and "Fehmarnbelt" is described in Experiments section 5 on page 64.

3.4 Conclusions of Section

In this section, the modified SOM algorithm for marine vessel movement data classification into normal and abnormal classes is presented with possible retraining strategies.

The SOM method modification is achieved by incorporating virtual phe-romone intensity calculation at the last epoch of model training. Further, during the model validation stage, the pheromone intensity threshold is introduced by applying a gradient descent method. The possibility to apply different neighbouring functions was depicted as well. With the view to decrease the computational time, different strategies for the retraining of the SOM network were developed and presented for further performance investigation. The data batching strategies with history data embedding allow the algorithm to cope with the huge amount of new data on vessel movements in a reasonable time.

4 Unsupervised Detection of Marine Vessel Abnormal Trajectory

This section presents the proposed algorithms and methods to achieve marine vessel unsupervised abnormal trajectory detection. The section is based on the literature review performed in section 1. The methods and algorithms related to the content of the subsequent section were published in papers [35, 81, 57, 82].

4.1 Marine Vessel Trajectory Prediction

For vessel trajectory unsupervised prediction the the deep neural network is applied. The deep neural network input is the previous specific vessel's navigational trajectory data, then the prediction of the vessel's subsequent position is calculated by the algorithm. If the prediction obtained by the algorithm falls within the defined limit, the vessel's expected location is considered as normal, otherwise it is considered abnormal.

Long short term memory neural network. Fully connected dense artificial neural networks do not ensure history retrospective. This is a significant drawback when forecasting algorithms try to predict time-series data in such domains as economics, language processing or transport routes. Recurrent Neural Networks (RNN) were proposed to overcome this challenge. However, RNN have vanishing and exploding gradients problems, and to overcome this in practice when longer sequences of input data are used, the dissertation proposes to use Long Short Term Memory (LSTM) network [83]. The network performs significantly better in other applications such as speech recognition [84], handwriting recognition [85], reinforcement learning [86] and many other fields.

LSTM structure implements modified back-propagation approach of gradient descent method that solves vanishing gradient problem and the network can learn complex non-linear patterns. LSTM network architecture represents interconnected cells and is shown in Figure 12. LSTM cells transmit cell state $c(t)$ (see Figure 12), that is passed to a network with minimal linear operations. Such passed information is often called LSTM cell memory. $h(t)$ is hidden state of cell and it is the same as cell output $y(t)$. This cell receives hidden state, $c(t-1)$ cell state, and $x(t)$ cell input from previous cell $h(t-1)$. Then the cell computes what information should be kept for further calculations and what has to be forgotten [83]. The architecture of LSTM auto-encoder can be obtained by interconnecting such cells.

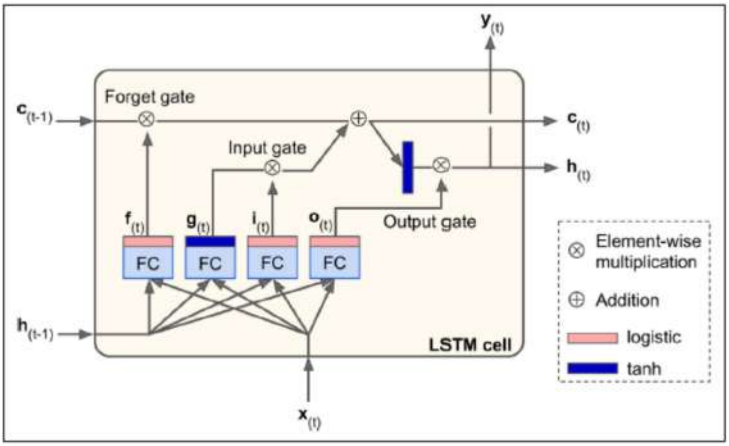


Figure 12: Structure of LSTM cell [83]

LSTM auto-encoder. An auto-encoder is a type of artificial neural network used to learn efficient data encoding in an unsupervised manner [87]. The goal of an auto-encoder is to extract a representational latent vector (encoding) for a set of data. A typical auto-encoder consists of three parts, namely, encoder, latent vector, and decoder. During training, encoder and decoder learn to reduce input and to reconstruct output through compressed latent vector in such way that the network input would be as close as possible to the network output. The main difference of the LSTM auto-encoder is that the main blocks of the network architectures are LSTM cells. Encoder compresses input data χ (see eq. (6)) to latent space and decoder predicts sequence of next vessel positions Y (see eq. (6)). In this dissertation a multivariate multi-step LSTM auto-encoder is used (see Figure 13). The decision on selection of Deep neural network architecture was made based on research by Jurkus presented in thesis [88], where best accuracy of vessel trajectory was archived with a multivariate multi-step LSTM auto-encoder.

The main parts of a proposed LSTM auto-encoder are input layer, encoder layers, a vector of encoded latent representation, decoder layers, and an output/reconstructed sequence layer. The input layer receives structured navigational vectors sequences χ defined by (6) and returns the predicted/reconstructed output sequences $\hat{Y}$ defined by (6). Encoder and decoder parts consist of LSTM cells that are interconnected sequentially inside a particular block and are parallel between on-top-stacked layers.

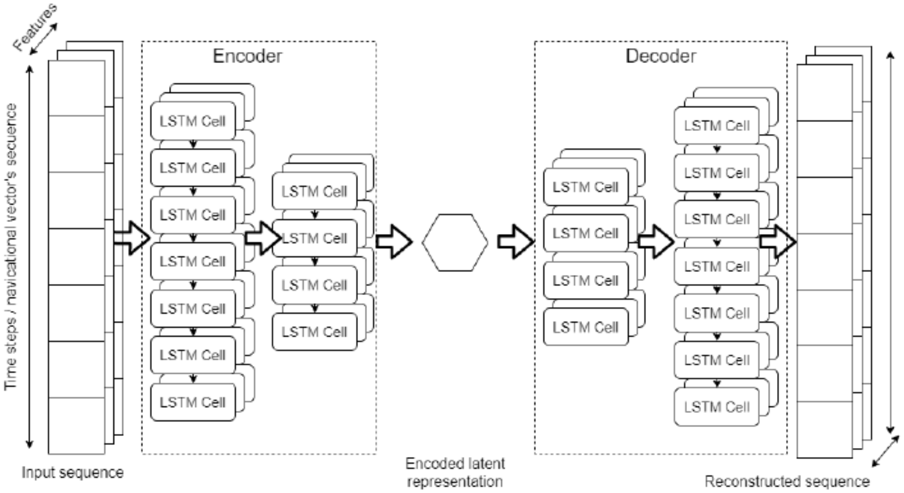

Features
Time steps / navicational vector's secuence
Input sequence
Encoder
LSTM Cell
Encoded latent representation
Decoder
Reconstructed sequence

Figure 13: Architecture of LSTM auto-encoder

A reconstruction (prediction) error of auto-encoder is obtained by:

$$e^{(l)}_{(g,r,j)} = Y_{(g,r,j)} - \hat{Y}^{(l)}_{(g,r,j)};$$

$$\begin{aligned} l \in \{upper, crisp, lower\}; \quad g \in \{1, 2, \ldots, N\}; \\ r \in \{1, 2, \ldots, \widetilde{n}\}; \quad j \in \{1, 2, \ldots, f\}; \end{aligned} \tag{11}$$

where $e^{(l)}_{(g,r,j)}$ is the reconstruction error of a single navigational vector's feature, $Y_{(g,r,j)}$ - true value of a single navigational vector feature, $\hat{Y}^{(l)}_{(g,r,j)}$ - output estimated value by auto-encoder of a single navigational vector feature, l is model type: crisp, lower, upper (described in section bellow 4.2), g - position of sequence of predicted navigational vector, r - position in sequence of predicted navigational vector, $\widetilde{n}$ - output sequence length, j - j^{th} navigational vector's feature, f - number of features, Y and $\hat{Y}$ are navigational vectors structured by expression (6). The loss function defined in equation 12 is used as is in crisp type of a model. The upper and lower type of models modify the equation 12 to obtain bound properties of the crisp model (see subsection 4.2).

$$L^{(l)}_s = \frac{1}{N\widetilde{n}f} \sum_{g=1}^{N} \sum_{r=1}^{\widetilde{n}} \sum_{j=1}^{f} (e^{(l)}_{(g,r,j)})^2, \quad l = \{upper, crisp, lower\}, \tag{12}$$

where L^l_s is the shared part of loss function for l (upper, crisp, or lower) type of models, N - number of training sequences in the training data set, s is the index that notes shared part of loss function expression between crisp, lower and upper type models.

4.2 LSTM Prediction Region Learning

Longitude and latitude coordinates are used to determine the abnormal vessel traffic. Real vessel navigational vectors are compared with predicted ones by the model in two-dimensional space. The assumption is, if the vessel's true position vector lays outside of the prediction region (multivariate case of prediction interval), it is interpreted as abnormal vessel movement and all vessel traffic vectors that appear inside the prediction region are interpreted as normal vessel movement.

Three differently configured LSTM auto-encoders are used to learn the models of multivariate prediction region boundaries. The Crisp ($l = \{crisp\}$) model type predicts the geographical coordinates of vessel trajectory. The lower ($l = \{lower\}$) model type predicts a lower bound of prediction region for Crisp type model. The upper ($l = \{upper\}$) model type predicts an upper bound of the prediction region for the crisp type model. Lower and upper bounds models predict the prediction region for the crisp type model.

In typical configuration, LSTM auto-encoder predicts only most accurate values (crisp). In order to determine prediction regions, a method was proposed by N.Cruz *et al.* [52] is used with modification to support multivariate and multi-step type LSTM networks. The prediction region is composed of upper and lower bounds in which the prediction/reconstruction output is found with a certain probability α [89]. The region is learned by training two LSTM auto-encoders with combined classical MSE loss function (12) with the second metric of region loss function as presented in [90]. The specific loss function for upper and lower bounds is defined as follows:

$$L_{\ell}^{(upper)} = \frac{1}{N\widetilde{n}f}\sum_{g=1}^{N}\sum_{r=1}^{\widetilde{n}}\sum_{j=1}^{f}(ReLU(e_{(g,r,j)}^{(upper)}))^2, \tag{13}$$

$$L_{\ell}^{(lower)} = \frac{1}{N\widetilde{n}f}\sum_{g=1}^{N}\sum_{r=1}^{\widetilde{n}}\sum_{j=1}^{f}(ReLU(-e_{(g,r,j)}^{(upper)}))^2, \tag{14}$$

where $L_{\ell}^{(upper)}$ and $L_{\ell}^{(lower)}$ are specific loss functions for upper and lower bounds respectively, ℓ - is index that notes specific part of loss function expression, *ReLU* is the rectified linear unit function defined by:

$$ReLU(x) = \begin{cases} 0, & \text{for} \quad x < 0 \\ x, & \text{for} \quad x \geq 0. \end{cases} \tag{15}$$

As presented in paper [90], data points $Y_{(g,r,j)}$ larger than $L_{\ell}^{(upper)}$ apply a cost equivalent to the squared difference between the real data point and its upper bound prediction/reconstruction in accordance eq. (13). Likewise, data points $Y_{(g,r,j)}$ lower than $L_{\ell}^{(lower)}$ are penalized as defined in equation (14). Data points $Y_{(g,r,j)}$ that are in prediction region (below upper and above lover bounds) have no cost with a help of ReLU function (15).

In combination of the upper and lower loss functions, a higher loss value is applied to $Y_{(g,r,j)}$ points that are outside of the prediction region. These regions are learnt by using the same target input data during training process. The overall loss function is defined as the weighted sum of the MSE (12) and the region loss functions (14)(13) for upper and lower bounds respectively [90][52]:

$$L_{total}^{(upper)} = L_{s}^{(upper)} + \lambda L_{\ell}^{(upper)}, \tag{16}$$

$$L_{total}^{(lower)} = L_{s}^{(lower)} + \lambda L_{\ell}^{(lower)}, \tag{17}$$

where $L_{total}^{(upper)}$ is overall upper loss function, $L_{total}^{(lower)}$ is overall lower loss function, λ is a tuneable parameter that represents the relative importance of the proposed classical/common and region loss functions [52]. The crisp model's output is learned by using only a MSE loss function (12):

$$L_{total}^{(crisp)} = L_{s}^{(crisp)}, \tag{18}$$

where $L_{total}^{(crisp)}$ is loss function for crisp model.

With these loss functions, the minimization of the prediction region area is achieved. If functions are not applied, the prediction region loss functions (L_i) increase the region area, introducing a trade off between the number of points that fall into the region and its area which can be regulated by modifying the parameter λ in eqs. (16) and (17). The algorithm for selection of λ is depicted in Figure 14

With the view to evaluate the quality of the prediction region, two indicators were used. The first is the prediction region coverage probability (PICP) that quantifies the number of measured values that fall within the region defined by the model [52] and is modified to support multi-variate features and multi-step predictions:

$$PICP = \frac{1}{N\widetilde{n}f} \sum_{g=1}^{N} \sum_{r=1}^{\widetilde{n}} \sum_{j=1}^{f} (\delta_{(g,r,j)}) \tag{19}$$

$$\delta_{(g,r,j)} = \begin{cases} 1 & \text{if} \quad Y_{(g,r,j)} \in \left[\hat{Y}_{(g,r,j)}^{(lower)}, \hat{Y}_{(g,r,j)}^{(upper)}\right] \\ 0 & \text{if} \qquad\qquad \text{otherwise.} \end{cases} \tag{20}$$

The second metric is Prediction Interval Normalized Average Width (PINAW) that is used to measure the area of the region [52]. PINAW was also modified for multi-step and multivariate features:

$$PINAW = \frac{1}{N\widetilde{n}fR}\sum_{g=1}^{N}\sum_{r=1}^{\widetilde{n}}\sum_{j=1}^{f}(\hat{Y}_{(g,r,j)}^{(upper)} - \hat{Y}_{(g,r,j)}^{(lower)}) \tag{21}$$

where R is the maximal difference between the feature $\max(\hat{Y}_{(g,r,j)}^{(upper)} - \hat{Y}_{(g,r,j)}^{(lower)})$ in the data set [52], [90].

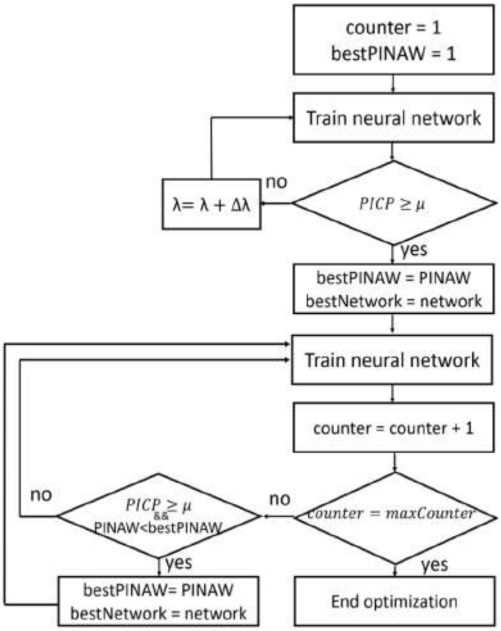


Figure 14: Iterative training process of joint supervision [90]

Figure 14 shows the algorithm for iterative training of the network. In this approach, the λ parameter is increased iteratively to force a wider region area in each iteration as the coverage probability increases. In each iteration the PICP is estimated by eq. (19) [90]. When the desired coverage probability α has been achieved, the algorithm stops the λ parameter incrementation. Few more iterations are calculated using fixed λ parameter in order to compensate the random initialization of the initial algorithm weights [90].

4.3 Wild Bootstrapping Prediction Region

One of the main advantages of the bootstrapping techniques is that it does not require to make any assumptions on the distribution of the data set being investigated. Traditionally bootstrap method re-samples the initial data to produce more data samples that could be used in repetitive experiment. However, the wild bootstrap technique is bit different. Instead of generating bootstrap samples that consist of re-sampling the original data or residuals, the wild bootstraps combine the data with random variables drawn from a known distribution to form a bootstrap sample. The usage in this dissertation can be summarised as:

1. Preparation of data as described in section 2.
2. Calculation of data set's variance for every feature type.
3. Generation of multi-variate normal random variables while keeping the same dimension and the mean equal to zero, and the variance the same as that of the input data.
4. Element-wise summation of the initial data set with the newly generated set, i. e. noise is added to the data with mean and variance calculated from initial data set.
5. Scaling of resulting data for better LSTM training results into interval $[0, 1]$ while keeping each feature scaling factors for predicted data reconstruction purpose.
6. Training of LSTM auto-encoder network.
7. Calculation of LSTM network predictions r-step ahead, $r \in \{1, 2, \ldots, \widetilde{n}\}$ ($\widetilde{n} = 50$).
8. Restoration of prediction scaling, i. e. up-scale predicted values according to the saved feature's scaling parameters described two steps above.
9. Repetition of steps 3-8 k-times (100 times in the dissertation experiment).

After the application of the scheme as proposed above, the matrix with predicted values is obtained. Then as point predicted value, the mean vector of k replicates is chosen for each feature and each prediction step. Thus $100(1-\alpha)\%$ prediction region for the mean (average predicted value) of a p-dimensional normal distribution is the ellipsoid determined for unknown μ such that (see [91]):

$$\frac{kr}{k+r}(\bar{x}_r - \mu)^T \hat{S}^{-1}(\bar{x}_r - \mu) \leq \frac{(k-1)p}{k-p} F_{p,k-p}(1-\alpha), \tag{22}$$

where

- $\bar{x}_r = \frac{1}{k}\sum_{u=1}^{k} x_{u,r,j}$ - the mean vector for each of the feature $j \in \{1 \dots f\}$ at each prediction step r,
- $\hat{S}$ - sample covariance matrix,
- $F_{p,k-p}(1-\alpha)$ is an $1-\alpha$-level critical value of a Fisher distribution with p and $k-p$ degrees of freedom.

4.4 Aggregation of Anomaly Detection Models

The previous subsections discussed the prediction regions learning, the LSTM prediction learning, and the LSTM wild bootstrapping methods. Initially, they were applied for single vessel type in a single level of prediction region. In previous researches [74], [73], [35] it was shown that vessel types have different and distinguishable traffic patterns are visible even in traffic visualisation of vessel types in the geographical plane (see Figure 6 37 p.). By inspecting visualization results, I have decided to create sets of model groups for each vessel type, where each set is responsible for a particular vessel's type abnormality detection. Moreover, each vessel type model set contains groups of models for individual prediction levels. Thus, aggregation of multiple prediction region models to detect a marine vessel traffic abnormality is described in this subsection.

Training of multiple models. Figure 15 shows the architecture of vessel trajectory prediction models for abnormal movement prediction. It depicts a process of training of the models and classification of unseen vessel navigational vectors as abnormal or normal.

The training part of aggregating model consist of these steps (see Figure 15):

1. The raw marine traffic data is collected from a Automatic Identification System (AIS) and meteorological data from a meteorological data provider.
2. The collected data is prepared as described in subsection 2. The prepared vessel navigational vector's sequences are separated to different sets by vessel type and stored as multiple data sets. Sequences that have no vessel type ("undefined") are stored separately for further preprocessing.
3. Data set with known vessel type has formed training set for vessel type recognition. The data set is formed in such a way that navigational vector sequences form input data, which is separated by vessel type. This separation by vessel type defines vessel type classes.

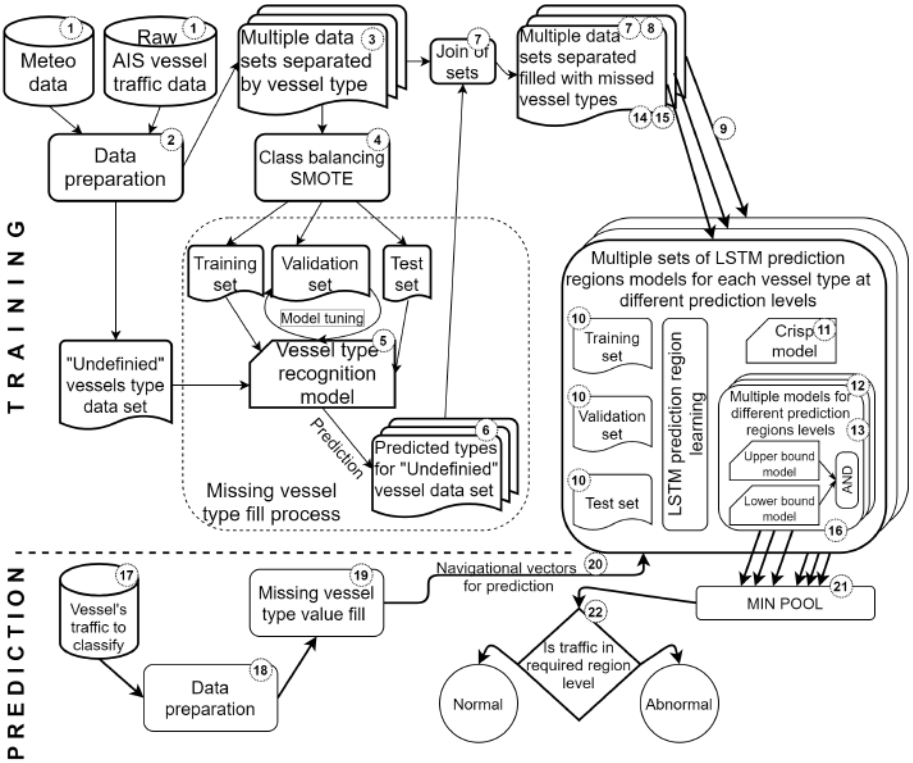

TRAINING
PREDICTION
Meteo data
Raw AIS vessel traffic data
Data preparation
Multiple data sets separated by vessel type
Class balancing SMOTE
Join of sets
Multiple data sets separated filled with missed vessel types
Training set
Validation set
Test set
Model tuning
Vessel type recognition model
"Undefinied" vessels type data set
Prediction
Predicted types for "Undefinied" vessel data set
Missing vessel type fill process
Multiple sets of LSTM prediction regions models for each vessel type at different prediction levels
Training set
Validation set
Test set
LSTM prediction region learning
Crisp model
Multiple models for different prediction regions levels
Upper bound model
Lower bound model
AND
Vessel's traffic to classify
Data preparation
Missing vessel type value fill
Navigational vectors for prediction
MIN POOL
Is traffic in required region level
Normal
Abnormal
1
1
2
3
4
5
6
7
7
8
9
10
10
10
11
12
13
14
15
16
17
18
19
20
21
22

Figure 15: Aggregation architecture of vessel trajectory prediction models for abnormal movement detection

4. Data set labels are balanced with Synthetic Minority Oversampling Technique (SMOTE) before performing a training process.

5. The Vessel type recognition model is trained as described in subsection 2.8. The trained model is saved for "undefined" vessel type classification.

6. The missing vessel type vectors are classified with earlier trained model. The new sets of data are constructed based on predicted vessel type.

7. The predicted data sets are joined with the rest of data sets. Multiple data sets are formed and grouped by vessel type.

8. All data set groups are marked as not processed.

9. A random data set of single vessel type is taken from the list of not processed data set groups.

10. Data set is split to training, validation and test subsets.

11. A crisp model is trained as described in subsection 4.2.

12. Multiple sets of Lower and Upper model pairs is trained with different λ, where $\lambda \in \{\lambda_{start}, \lambda_{start} + \Delta\lambda, ..., \lambda_{stop}\}$, and $\Delta\lambda$ is a step of incremental λ increase, λ_{start} and λ_{start} are start and end of λ incremental increase respectively. The training process is described in subsection 4.2. The trained model and it's weights are stored.

13. $PICP$ and $PINAW$ are calculated for each model pair and stored for further use.

14. The vessel type data set, that was used for training, is marked as processed.

15. If vessel type data set has not yet been processed, repeat from step 9.

16. All models are grouped by vessel type and then inside of a group each pair of upper and lower model is given a particular $PICP$ value. Models are stored and further used for classification of vessel traffic abnormality.

Each vessel type group has one crisp type model and number of upper/lower models pairs sets sub-grouped by $PICP$. All models with particular vessel types and $PICP$ information are stored. These multiple data sets are used for vessel movement classification for abnormality classification at predefined level of prediction region.

Classification of vessel trajectory abnormality. The process of classification is as follows:

17. The vessel navigational vectors are collected from Automatic Identification System (AIS) system. The minimum $\widetilde{n} + n'$ number at $\Delta T_{interval}$ time interval of sequential vessel navigation vectors is required.

18. The collected data are prepared as described in subsection 2.

19. If vessel type is unknown or "undefined", the Vessel type recognition model is used for vessel type prediction.

20. The prepared sequence of vessel navigational vectors is provided for a set of models for a particular vessel type. Then all upper and lower bound models are used to check that true values of vessel sequences are inside of the prediction region of particular model pair. If model predicts that true value is in prediction region, it

returns $PICP$ value for this model. The set of upper/lower model pairs returns a set of $PICP$ values.

21. The Min Pool layer [92] returns minimal value of $PICP$ provided by models.
22. The decision layer based on set of required $PICP$ threshold. Checks are performed whether vessel traffic is in required prediction region level. If it is not in required prediction level, the vessel traffic is classified as abnormal, otherwise it is considered normal.

4.5 Conclusions of Section

This section describes the LSTM prediction region learning and LSTM wild bootstrapping methods for anomalous vessel trajectory prediction. The dissertation assumes that if the vessel's actual trajectory coordinates are in a prediction region, its trajectory is normal. Otherwise, when it is outside the prediction region, it is classified as abnormal.

The proposed LSTM prediction learning method relies on three LSTM prediction region models, namely, crisp, lower and upper. A suggestion is provided for modification of loss function of the separate LSTM lower and upper bounds models to learn vessel's trajectory prediction regions.

In order to compare the results, the wild bootstrapping method and its integration to LSTM auto-encoder were used in the dissertation. The LSTM wild bootstrapping learns prediction regions based on the statistical technique. The technique mixes data with multivariate normally distributed random variables and performs this procedure multiple times during the training. After the numerous LSTM auto-encoders training, the method calculates prediction region ellipses for the required abnormal vessel detection.

5 Experiments and Results

This section presents a series of experiments intended to compare the performances of the proposed marine anomaly detection methods. The section describes the results of the application of unsupervised and semi-supervised anomaly detection algorithms, introduces anomalous trajectory grouping, and describes the strengths and weaknesses of the proposed methods.

Figure 16 shows the software development architecture for investigation of abnormal marine traffic methods. The first layer of the architecture contains computational infrastructure. The main high-performance computing (HPC) is performed on two dedicated Nvidia RTX2080i GPUs and Intel i7 core ×64 CPU. The first SSD 0.5 TB disk was dedicated for Linux Debian 10.0 OS. The second SSD 4 TB disk was used for storage of data sets and model data.

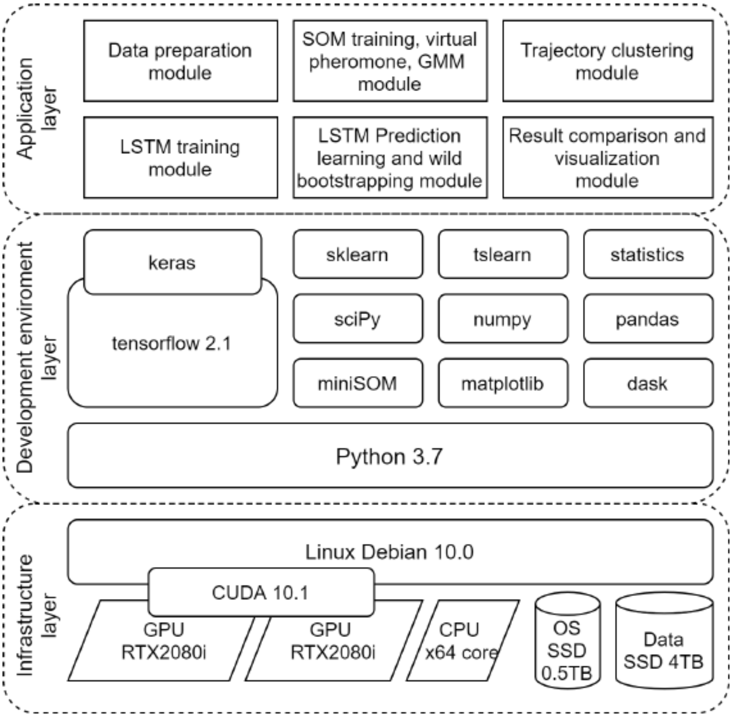


Figure 16: Software development architecture for abnormal marine traffic methods investigation

The second layer of the architecture was dedicated for software development environment. All development was done on Python v3.7 programming language/interpreter. Methods are implemented with the help of multiple libraries, tensorflow 2.1, keras, miniSOM, SciPy, tslearn, etc.

The third layer has a program that was developed to implement described methods in the thesis and code script for experiments. The layer has the following modules: data preparation, SOM training, virtual pheromone, GMM training, LSTM training, LSTM prediction region learning, wild bootstrapping, result comparison, and visualization modules. All intermediate training steps were stored as checkpoints and resumed in case of system failure or restart.

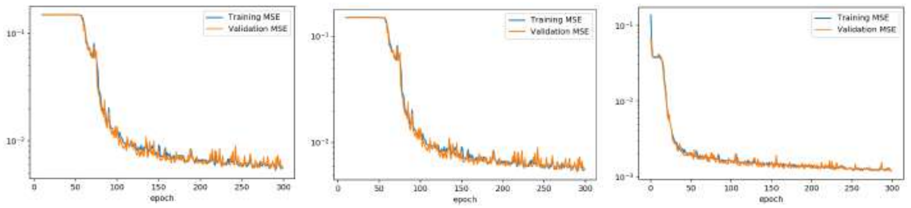
5.1 Performance of LSTM Prediction Region Learning for Detection of Anomalous Trajectories

In this subsection the results of LSTM prediction region learning method presented in 4.1, 4.2 on pages 53, 56 is evaluated. The subsection describes experimental investigation and the results obtained for each vessel type (further named by feature name: "Vessel type") type for crisp, upper, lower trained deep neural network models. The data was randomly split into training non-overlapping, validation, and test subsets (see Table 14).

Artificial neural network setup. LSTM prediction region training me-thod uses LSTM multi-stacked auto-encoder described in subsection 4.1 paragraph *"LSTM auto-encoder"*. The input shape of network setup has an input sequence length $n' = 50$, number of features $f = 15$ and batch size of 512. The first LSTM layer has 128 LSTM cell units for each feature. The second LSTM layer has 64 units. The encoded latent representation vector has 16 units. Third LSTM layer has 16 units and the last layer has 128 units. Output layer returns two dimensional multistep vector of size $\widetilde{n} = 50 \times f' = 2$. For the network output only spatial features latitude and longitude are used. For all LSTM cells the Hyperbolic Tangent Function (TANH) activation function were applied. To ensure cross validation and regularization, each epoch was validated against validation data set. 300 epochs are set for network training. As an optimiser the Adam algorithm [63] is used with calibrated parameters: learning rate $\alpha_{adam} = 0.001$, exponential decay rate for the first moment $\beta_1 = 0.9$, exponential decay rate for the second-moment $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The selection of the LSTM architecture was based on master thesis [88].

Three different types of LSTM auto-encoders are configured with different loss functions: crisp network was configured to use loss function (18); lower and upper bounds LSTM networks were configured to use loss functions (17) and (16) (see page 57).

Training LSTM prediction region models. All the models were trained by the algorithm described in subsections 4.1 and 4.4. For each vessel type a 10 separate crisp models were trained with different initial random weighs (in total 220 crisp models). Then the sets of the lower and the upper models are trained in advance. In order to train these sets the sequence of λ is created. Multiple sets with different λ value of lower upper models pairs is trained for vessel type. For each λ and vessel type, 10 models sets were created with random initial neuron weights. There is total of 5280 lower/upper models pairs, including 24 different values of λ, 22 vessel types, and 10 iterations with random initial weights for each combination.
Figures 17 show typical decay of training and validation losses function over epochs. It is worth to note that, typically, training of an upper and an lower models starts with slow progress and only after 60^{th} epoch loss starts to drop significantly. During training of crisp type model, the validation loss drops after 20^{th}. The constant drop of training/validation losses (Figure 17) shows that the training of models for LSTM prediction region learning can be performed less that 300 epochs. However, more thorough research is needed.



(a) Upper bound (b) Lower bound (c) Crisp

Figure 17: Training/validation Losses over Epochs on logarithmic scale

Evaluation of trajectory prediction with crisp model. As mentioned above, 10 crisp type models were trained for each Vessel type. Table 16 shows crisp model errors mean and standard deviation. This table contains only the main error types. Other error types and visualisations of multiple type errors can be found in appendix E on page 114. Errors are calculated on test data sets. It is observed that the errors of test data sets are not significantly larger. That shows that the model generalization is satisfactory for further use in marine traffic anomaly detection.
The crisp model errors form two groups, with more minor errors and more significant errors. Models of vessel type Anti-pollution, Cargo, Passenger, Tug, and others have smaller error values. Those vessel type models

Table 16: Crisp model errors on 10 randomly trained LSTM networks

	Sequences		MAE, km		RMSE, km		MAPE, %		MASE	
Vessel type	Train	Test	mean	std	mean	std	mean	std	mean	std
Anti-pollution*	1094	365	0.63	0.65	5.62	6.95	0.06	0.07	0.30	0.31
Cargo	75625	25209	2.43	0.11	4.67	1.05	0.19	0.01	0.07	0.00
Diving*	103	35	20.20	4.11	30.35	17.57	2.01	0.48	0.71	0.14
Dredging	4584	1528	2.81	0.39	4.62	2.05	0.22	0.03	0.08	0.01
Fishing	15748	5250	2.88	0.12	4.26	1.06	0.24	0.01	0.08	0.00
HSC*	32	11	10.36	3.78	16.07	14.38	0.92	0.39	0.25	0.09
Law_enforcement	4467	1489	3.71	0.14	6.13	1.73	0.31	0.01	0.10	0.00
Military	4743	1581	4.90	0.17	8.17	1.95	0.42	0.01	0.16	0.01
Passenger	47988	15996	1.73	0.10	3.68	1.04	0.13	0.01	0.05	0.00
Pilot	6352	2118	4.00	1.61	7.76	6.56	0.32	0.12	0.09	0.04
Pleasure	1965	655	2.70	1.50	6.88	6.35	0.22	0.12	0.12	0.07
Port_tender*	272	91	10.91	5.14	20.60	14.45	0.98	0.45	0.32	0.15
Reserved*	436	146	7.93	2.82	3.59	10.04	0.65	0.27	0.26	0.09
SAR	3704	1235	1.58	0.75	3.83	2.20	0.14	0.08	0.14	0.07
Sailing	6692	2231	1.27	0.32	2.36	1.12	0.11	0.04	0.07	0.02
Spare_1*	15	6	1.88	1.13	2.93	3.65	0.13	0.06	0.23	0.14
Tanker	22577	7526	2.78	0.17	5.00	1.32	0.22	0.02	0.08	0.00
Towing*	522	175	17.40	8.90	24.65	25.38	1.52	0.95	0.46	0.24
Towing_long_wide*	367	123	8.00	3.51	10.23	12.68	0.65	0.34	0.20	0.09
Tug	9421	3141	2.01	0.35	3.67	1.65	0.16	0.03	0.05	0.01
WIG*	40	14	19.83	8.69	28.44	22.49	1.95	0.90	0.51	0.23
Total average			6.19	2.12	9.69	7.41	0.55	0.21	0.21	0.08
Average*			11.14	4.40	16.53	14.36	1.03	0.45	0.38	0.17
Average			2.73	0.48	5.09	2.34	0.22	0.04	0.09	0.02

Smaller data sets are marked with star ("*").

were trained with a more significant number of navigational vector sequences (N) compared against the group of models with more significant errors such as Diving, HSC, Towing, WIG, and others (see Table 16 and Figure 18). Figure 18 depicts that the same group with more significant errors also has more considerable error variance when the model is trained with different random initial weights. Similar dependency is visible in other types of errors observed in other error types visualizations (Appendix E).

Prediction region: upper and lower models. As mentioned above, multiple upper and lower model pairs are trained with different λ, where $\lambda_{start} = 5, \Delta\lambda = 5, \lambda_{stop} = 120$. Additionally $1, 2, 3, 4$ of λ values to investigate PICP and PINAW grow on smaller λ values. For all sets of λ's separate neural networks were trained for each vessel type. Each upper/lower model pair was trained 10 times to investigate the impact of random initial random network weights. For each vessel type, two

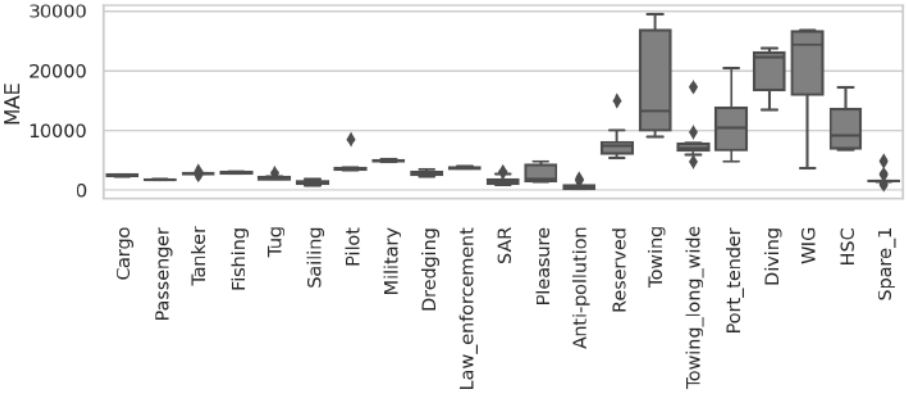
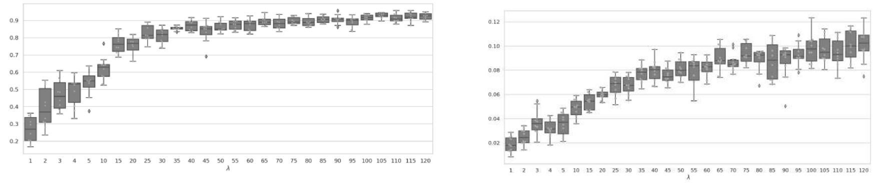


Figure 18: MAE errors distributions for different vessel types



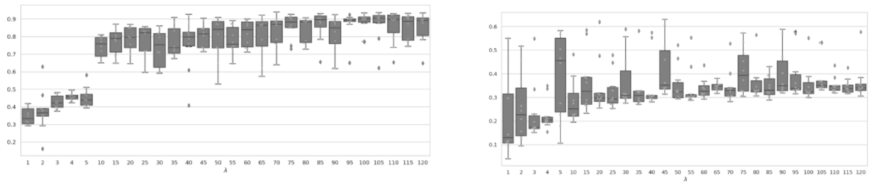
(a) PICP trend by λ

(b) PINAW trend by λ

Figure 19: PICP and PINAW trends of "Cargo" vessel type

distribution box charts were drawn, one for PICP trend over λ and second PINAW over λ (Appendix F, page 116). Two vessel type charts are shown as examples in Figures 19, 20. Mostly PICP grows in logarithmic manner to value 1, then λ stays linear. The variance of PICP for particular λ varies between vessel types and iterations with different initial neuron weights. This behaviour suggests that for model anomaly with particular PICP value, we must search corresponding λ and initial neuron weights. PINAW behaves in a similar way: it grows logarithmically with growing λ, that shows that prediction region covers larger area. This fact was taken in account in LSTM prediction learning algorithm described in subsection 4.2 on page 56.

Regarding all vessel types PICPs growths over λ (Appendix F, 116 p.), it was observed that majority of vessel types exceed 0.8 PICPs as λ values vary from 25 to 30. Vessel types such as Diving, HSC, Port_Tender, Spare_1, and Wig reach the same PICP value with much higher λ values, and Port_Tender does not reach 0.8. In addition, it was observed that the same vessel types have significantly fewer training sequences. The Figures 19, 20 depict Cargo and Diving vessels PICP and PINAW relations to λ with 10 iterations with random initial weights.



(a) PICP trend by λ

(b) PINAW trend by λ

Figure 20: PICP and PINAW trends of "Diving" vessel type

Cargo and Diving PICP growths represent samples of two types of PICP trends. Cargo (Figure 19a) represents a group of PICPs that grow more stable with smaller variance. Also, this group usually has a larger number of training sequences. The Diving (Figure 20a) represents a group where PICPs grow less stable with larger variance and outliers. The number of the training sequence in this group is significantly smaller.

Appendix F also shows growth of PINAW over λ. As λ increases, the PINAWs grows logarithmically. The PINAW trend figures show that with higher training samples, the PINAW grows more stable (Figure 19b) compared to significantly smaller training sequences such as Diving (Figure 20b).

Discussed PICP and PINAW trends clearly show dependency on λ parameter and can be used for prediction region training in LSTM autoencoder neural network.

PICP and PINAW adjustment. The previous paragraph shows that the PICP and PINAW can be used to obtain the desired anomaly level $(1-\alpha)$. Subsection 4.2 on page 56 and Figure 14 on page 58 p. describes the algorithm for corresponding PICP searching. PICP search results for anomaly level $(1-\alpha), \alpha = 0.05$ is summarized in the Table 17. Percentage values instead of ratios was chosen for better readability. For each vessel type the closest PICPs at particular λ for $(1-\alpha)$ were found. The searches were performed on the validation data set and then tested on the test data set. The tested PICPs have similar values as those obtained in validation data set that show good generalization of the models. However, it is observed that vessel types such as Diving, HSC, Port_tender, Spare_1, WIG with smaller training sequences did not reach the desired PICP value.

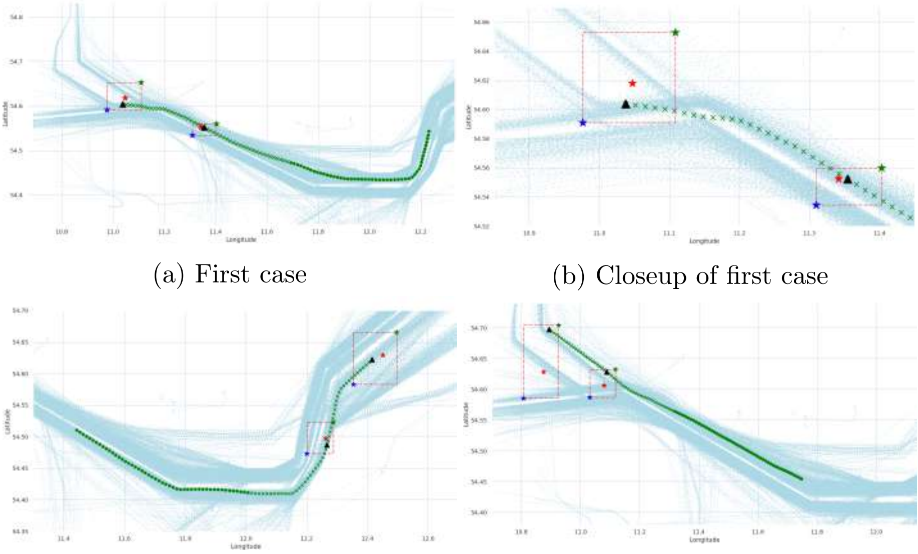
Anomaly detection. Trained models were tested on the test data set. Figure 21 shows randomly selected cases of normal vessel trajectory. The figure depicts the vessel trajectory in the longitude and latitude plane,

Table 17: PICP values search results for $100(1-\alpha) = 95.0\%$ anomaly level

				PICP, %		PINAW, ratio $\times 10^3$	
	Sequences			on data sets		on data sets	
Vessel type	Train	Test	λ	validation	test	validation	test
Anti-pollution	1094	365	30	95.1	94.1	1.17	1.06
Cargo	75625	25209	120	95.0	96.0	0.07	0.07
Diving	103	35	70	94.0	83.5	9.78	9.92
Dredging	4584	1528	105	95.0	95.1	0.02	0.01
Fishing	15748	5250	115	94.9	94.4	1.09	1.07
HSC	32	11	50	83.7	84.6	112.79	110.80
Law_enforcement	4467	1489	80	95.3	95.5	3.32	3.63
Military	4743	1581	100	94.8	95.3	1.75	1.88
Passenger	47988	15996	105	95.2	95.9	2.09	2.06
Pilot	6352	2118	120	95.0	95.9	0.41	0.44
Pleasure	1965	655	110	95.1	94.3	1.15	1.21
Port_tender	272	91	120	48.5	48.7	465.15	507.79
Reserved	436	146	105	94.5	94.7	4.96	4.65
SAR	3704	1235	10	95.0	95.9	0.24	0.25
Sailing	6692	2231	75	94.9	95.2	0.94	1.01
Spare_1	15	6	75	91.3	90.3	37.50	34.39
Tanker	22577	7526	80	95.1	94.6	1.23	1.20
Towing	522	175	90	93.6	94.4	13.57	14.41
Towing_long_wide	367	123	120	93.5	92.5	14.80	15.56
Tug	9421	3141	50	95.0	95.5	0.14	0.13
WIG	40	14	85	87.7	87.4	72.64	72.34

Light blue color shows actual marine traffic (10000 trajectories) in the area, Green dots represent vessel navigational vectors used for model training input χ . Green crosses mark true values Y of vessel movement. Black triangles show true values of vessel positions $Y_{(g,25,j)}, Y_{(g,50,j)}$ at time frames of 25 and 50 respectively. Red star represent crisp model prediction $\hat{Y}^{(crisp)}_{(g,25,j)}, \hat{Y}^{(crisp)}_{(g,50,j)}$ at time steps 25 and 50. Green star shows prediction $\hat{Y}^{(upper)}_{(g,25,j)}, \hat{Y}^{(upper)}_{(g,50,j)}$ of upper bound model at time steps 25, 52 and Blue star marks prediction $\hat{Y}^{(lower)}_{(g,25,j)}, \hat{Y}^{(lower)}_{(g,50,j))}$ obtained by lower bound model at time steps 25, 50. Red dashed lines depict boundaries of the prediction region for a particular movement trajectory.

Figure 21 shows a few cases of vessel normal movement trajectories, which were passed into model and visualised in two dimensional latitude/longitude planes. The model was only introduced to a sequence of



(a) First case

(b) Closeup of first case

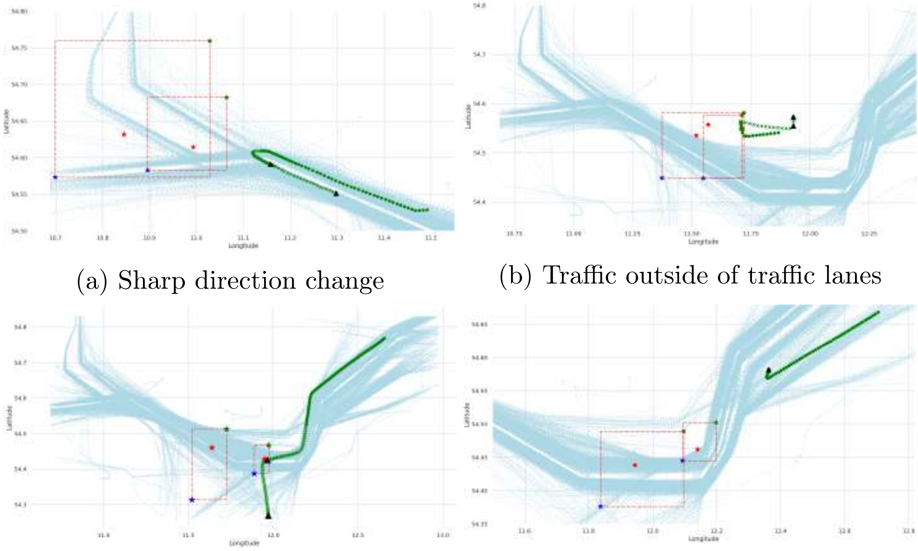
(c) Second case

(d) Third case

Figure 21: Random cases of <u>normal</u> vessel traffic

first 50 vessel navigational vectors, that are marked as green dots (seen in Figures 21a, 21d, 21d). Green crosses mark sub-sequential vessel positions Y, which were unknown to the model. Model predicts upper $\hat{Y}^{(upper)}$ and lower bounds $\hat{Y}^{(upper)}_{(g,25,j)}$, shown in green, blue stars and red doted rectangles. True value lays inside the rectangle and the vessel trajectory should be considered as normal according to LSTM prediction model. This situation illustrates the case when the LSTM prediction region method indicates traffic as normal. It is observed that the model in narrow marine traffic area learns the smaller prediction regions and vice versa. This is seen in Figure 21d, where the first prediction region is smaller because it is on junction of vessel routes. And, if vessel routes split up, the prediction region becomes wider by covering almost all possible routes of the specific vessel type.

Figures 22 depict abnormal vessel traffic cases (LSTM prediction region method has classified those as abnormal), but the Figure 22a shows anomalous cases: the cargo vessel unexpectedly turned around by changing direction 180 degrees due the captains decision to return to port for engine repairs. The first 50 navigational vectors of the vessel were used as model input, and the model predicted regions where the true position of the vessel is expected. Because vessel made sharp change in direction, the true values were outside of the predicted regions by both methods. The same Figure 22a shows that true 25^{th} and 50^{th} vessel positions (black triangles) are outside the prediction regions by LSTM region prediction



(a) Sharp direction change

(b) Traffic outside of traffic lanes

(c) Direction change

(d) Unexpected stop

Figure 22: Different types of abnormal vessel traffic

method (red rectangles). Thus, such vessel movement can be classified as abnormal, because it doesn't fall in $(1 - \alpha) = 0.95, PICP = 0.95$ prediction region. Figure 22b depicts a drifting cargo vessel, due to broken engine. Its actual navigational vectors are outside of the prediction region as well. The Figures 22c and 22d show few other anomalous movement cases: the (Figure c) shows an unexpected turn to a minor port and (Figure d) shows an unplanned stop due to the engine failure.

5.2 Performance of LSTM Wild Bootstrapping Technique for Detection of Anomalous Trajectories

This subsection describes the results of LSTM wild bootstrapping method application. The method is presented in subsections 4.3 on page 58. The method is used to train LSTM auto-encoders (subsection 4.1 on page 53 for each vessel type by using training data sets and afterwards are tested using test data sets (see Table 14). The neural network hyper parameters were set the same as described in LSTM prediction region part listed in subsection 5.1 in paragraph *Configuration of artificial neural network* on page 65.

Wild bootstrapping for trajectory prediction. For each vessel type a separate set of models are trained for trajectory prediction. The training data is taken from prepared and cleaned data sets (see Table 14).

The wild bootstrapping is applied to each vessel type separately. Then the $k = 100$ LSTM auto-encoders are created and trained in a following way: multi-variate normal random variables are generated while keeping the same dimensions and the means equal to zero, and the variances the same as that of the input data χ $(n' = 50, f = 15, N)$; the number of sequences N for each vessel type can be found in Table 14; the generated normal variables are summed element wise with χ; new summed input and expected output was Y scaled to [0.1] interval; the LSTM auto-encoder was trained with 300 epochs; LSTM network predictions were calculated r-step ahead, where $r \in 1, 2, ..., m$, $m = \widetilde{n} = 50$ with training and test separately; the scaling is restored; these steps were repeated for all k LSTM auto-encoders. The method in detail is described in subsection 4.3 on page 58.

Trajectory prediction accuracy. The method requires to train 2200 LSTM neural networks. 22 vessel types × k, where $k = 100$. The wild bootstrapped prediction $\bar{x}_r$ is calculated as described in equation (22). Table 18 contains prediction MAE, RMSE, MAPE, and MASE errors of $(Y - \bar{x}_r)$ for each vessel type. The prediction errors are calculated using train and test data sets. Errors of test data sets are not significant compared to the errors of the training data set, which shows that model generalization is acceptable for further use in marine traffic anomaly detection. Results show that the trajectory prediction models form two groups with more minor errors and more significant errors. Models of vessel type Anti-pollution, Cargo, Passenger, Tug, and others have minor errors. The same group has larger training data sets compared to a group of models with more significant errors such as Diving, HSC, Towing, WIG, and others (see Table 18). The LSTM prediction region model errors have similar behavior, where the same vessel type with the small count of training sequences has more significant prediction errors.

Evaluation of PICP for LSTM wild bootstrapping. After LSTM wild bootstrapping models were trained and prediction errors evaluated, the PICPs are calculated for the prediction level $100(1 - \alpha) = 95.00\%$. Table 19 contains results of PICPs for each vessel type of the training and test data sets. The table shows that vessel types such as Cargo, Dredging, Fishing, Law_enforcement, Military, Passenger, Pilot, Pleasure, Pilot, Pleasure, SAR, Sailing, Tanker, and Tug have PICP value nearly the same as prediction region (1-α). This behavior relates to the fact that these vessel types possess larger training sequences compared to other vessels. Vessel types such as Anti-pollution, Diving, HSC, Port_tender, Reserved, Towing, Towing_long_wide, and WIG have sig-

Table 18: LSTM Wild Bootstrapping prediction ($Y - \bar{x}_r$) errors

	Sequences		MAE, km		RMSE, km		MAPE, %		MASE	
Vessel type	Train	Test	train	test	train	test	train	test	train	test
Anti-pollution*	1094	365	1.08	1.08	5.89	5.89	0.23	0.23	1.04	1.04
Cargo	75625	25209	1.74	1.74	3.13	3.13	0.27	0.27	0.09	0.09
Diving*	103	35	11.55	11.62	16.85	16.82	2.35	2.34	0.81	0.81
Dredging	4584	1528	1.78	1.78	3.03	3.03	0.29	0.29	0.10	0.10
Fishing	15748	5250	1.93	1.94	2.76	2.75	0.33	0.33	0.10	0.10
HSC*	32	11	16.33	16.35	21.90	21.96	3.25	3.24	0.78	0.79
Law_enforcement	4467	1489	2.35	2.36	3.68	3.69	0.40	0.40	0.13	0.13
Military	4743	1581	2.95	2.95	4.85	4.88	0.52	0.52	0.19	0.19
Passenger	47988	15996	1.31	1.31	2.38	2.38	0.19	0.19	0.08	0.08
Pilot	6352	2118	1.85	1.85	3.60	3.61	0.31	0.31	0.08	0.09
Pleasure	1965	655	5.59	5.62	9.33	9.35	1.09	1.09	0.49	0.49
Port_tender*	272	91	14.83	14.90	19.47	19.54	2.43	2.44	0.88	0.88
Reserved*	436	146	11.09	11.13	17.00	17.09	2.18	2.18	0.71	0.72
SAR	3704	1235	0.91	0.91	3.20	3.20	0.16	0.16	0.16	0.16
Sailing	6692	2231	0.85	0.85	1.59	1.59	0.15	0.15	0.09	0.09
Spare_1*	15	6	2.44	2.45	3.40	3.41	0.26	0.26	0.61	0.61
Tanker	22577	7526	1.88	1.88	3.28	3.28	0.30	0.30	0.11	0.11
Towing*	522	175	14.74	14.78	20.00	20.10	2.83	2.83	0.78	0.79
Towing_long_wide*	367	123	14.03	14.04	19.96	20.05	2.80	2.80	0.71	0.71
Tug	9421	3141	1.67	1.67	2.75	2.75	0.29	0.29	0.08	0.08
WIG*	40	14	13.32	13.34	18.52	18.49	2.67	2.68	0.69	0.69
Total average			5.92	5.93	8.88	8.90	1.11	1.11	0.41	0.42
Average*			10.67	10.71	15.38	15.41	2.03	2.03	0.79	0.79
Average			2.07	2.07	3.63	3.64	0.36	0.36	0.14	0.14

Smaller data sets are marked with star ("*").

nificantly smaller PICP value than the desired prediction region (1-α). The same dependency of the PICP dynamics was observed in the LSTM prediction region learning experiment (the results are shown in table 17).

Anomaly detection. All trained models were tested using the test data set. Figure 23a shows randomly selected cases of normal vessel trajectories. The figure depicts the vessel trajectories in longitude and latitude plane. Light blue color shows actual marine traffic (10000 trajectories) in the area. Green dots represent vessel navigational vectors used for model training input X. Green crosses mark true values Y of vessel movement. Black triangles show true values of vessel positions Y_{25}, Y_{50} at time steps 25 and 50 respectively. Red stars represent LSTM

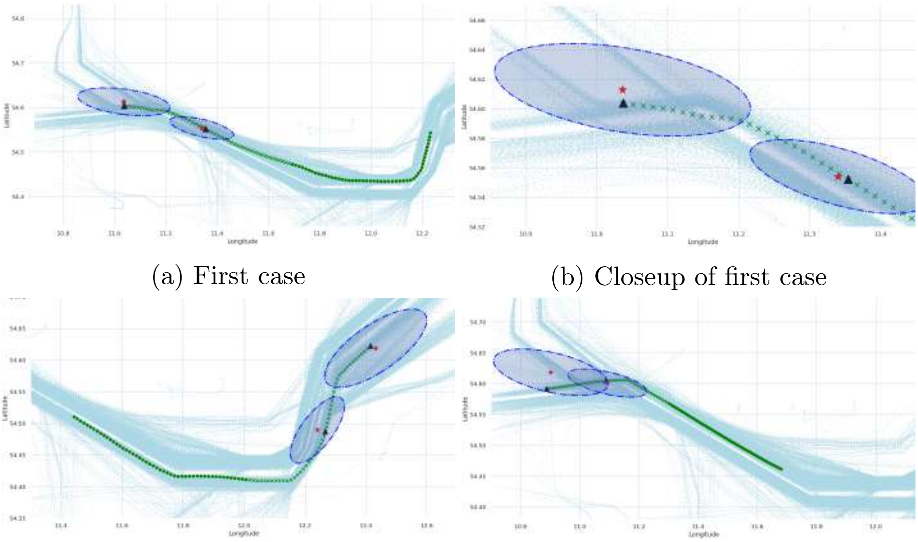
Table 19: PICPs values search results for $100(1-\alpha) = 95.00\%$ prediction region

	Sequences		$PICP, \%$	
Vessel type	Train	Test	train	test
Anti-pollution	1094	365	49.11	49.14
Cargo	75625	25209	94.78	97.51
Diving	103	35	0.01	0.00
Dredging	4584	1528	89.30	89.21
Fishing	15748	5250	94.34	89.66
HSC	32	11	3.49	0.91
Law_enforcement	4467	1489	81.30	81.62
Military	4743	1581	82.72	82.36
Passenger	47988	15996	96.59	96.39
Pilot	6352	2118	82.69	82.89
Pleasure	1965	655	87.21	86.81
Port_tender	272	91	0.01	0.00
Reserved	436	146	0.07	0.02
SAR	3704	1235	95.90	96.15
Sailing	6692	2231	97.93	98.14
Spare_1	15	6	27.50	7.42
Tanker	22577	7526	96.42	96.84
Towing	522	175	1.55	0.01
Towing_long_wide	367	123	0.61	0.01
Tug	9421	3141	94.55	94.54
WIG	40	14	1.11	0.00

Wild bootstrapping model prediction $\bar{x}_{25}, \bar{x}_{50}$ at time steps 25 and 50. A blue dashed ellipse depicts boundaries of the prediction region for particular movement forecast. The elliptical region is calculated by the equation (22). The traffic that is in this region is defined as normal and the traffic outside of the region is assumed to be anomalous.

Figure 23 shows a few cases of vessel normal movement trajectories. The model was introduced to a sequence of first 50 vessel navigational vectors χ, that are marked as green dots (see Figures 23a, 23c, 23d). Green crosses mark sub-sequential vessel positions Y, which were unknown for the model. The model calculates prediction region ellipses with centers at $\bar{x}_{25}$ and $\bar{x}_{50}$, shown as blue dotted ellipse. True values are inside the blue ellipse, and the vessel trajectory should be considered as normal according to LSTM wild bootstrapping model.

Figure 24 a, b, c, and d depict abnormal vessel traffic cases. Figure 24a shows the anomalous case where a cargo vessel unexpectedly turned around by changing direction 180 degrees due captain's decision to return port to repair an engine malfunction. The first 50 navigational



(a) First case

(b) Closeup of first case

(c) Second case

(d) Third case

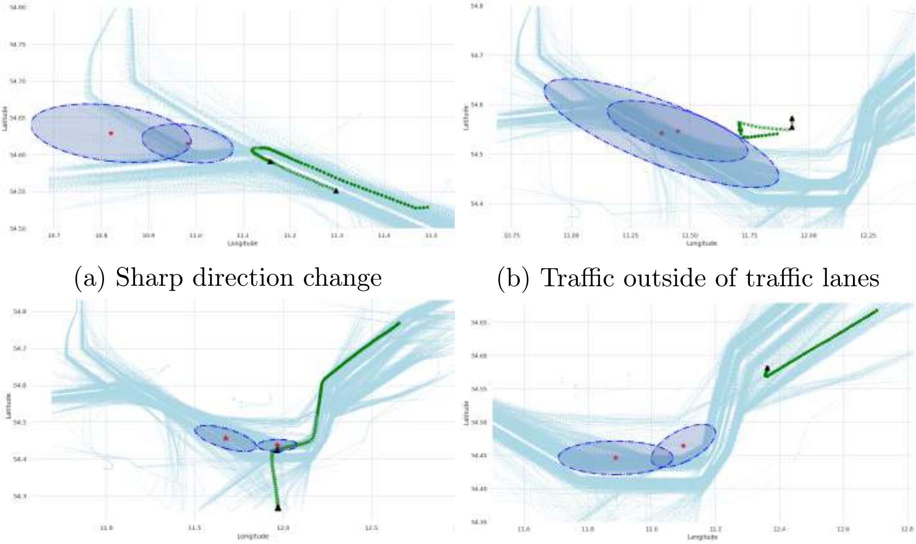
Figure 23: Random cases of normal vessel traffic

vectors of the vessel are fed to the input of the model. The model had predicted – the same case as it was described by LSTM prediction region learning – regions where the true position of the vessel is expected. Because the vessel made a sharp change in direction, the true values were outside of the predicted regions created by both methods. The same Figure 24a shows that true 25^{th} and 50^{th} vessel positions (black triangles) are outside of the prediction regions created by LSTM wild bootstrapping method (blue ellipse). The method classifies such vessel traffic as anomalous, because it doesn't fall in $(1-\alpha) = 0.95, PICP = 0.9751$ prediction region. Figure 24b depicts a drifting Cargo vessel because of a broken engine: the actual navigational vectors of the vessel are outside of the prediction region. Figures 24c and 24d show other abnormal cases as well: the figures depict an unexpected turn to a minor port and unplanned stop due to engine failure.

5.3 Performance of Point-based Method for Detection of Anomalous Trajectories

In this subsection, the experiments and results of SOM_pheromone and SOM_GMM algorithms are presented. The mentioned algorithms were described in section 3 *Point based detection of vessel traffic anomalies* on page 43.

Both point based methods were analysed using two different data sets: Klaipeda seaport area AIS data set [26, 27] (see Table 20) and



(a) Sharp direction change (b) Traffic outside of traffic lanes

(c) Direction change (d) Unexpected stop

Figure 24: Different types of abnormal vessel traffic

sea area "Fehmarnbelt" [53] described in section 2 *Data preparation* on page 18 (see Table 14, p. 41). The main differences between these data sets are the intensity and complexity of the sea traffic and subsequently the amount of navigational vectors to be analysed by the algorithms. After data cleaning, there are 138242 Klaipeda Cargo type vessels vs. 12604200 (126042 sequences) of Fehmarnbelt Cargo vessels. Similar situation is with other vessel types as well. Another important aspect is that Klaipeda's data set is annotated by the experts. This annotation was used to fine tune the SOM_pheromone β_{PPV}, β_{TPR} and SOM_GMM $P(H = normal)$ parameters in order to maximize accuracy of anomaly detection. On the other hand the "Fehmarnbelt" data set is huge. After data preparation, it has a total of 34459100 navigational vectors, which belong to all vessel types. It becomes obvious that data annotation by human expert is virtually impossible. Because of this and in order to study semi-supervised point-wise (SOM) methods using Fehmarnbelt

Table 20: Klaipeda seaport data set

	Navigational vectors		
Data subset	Total	Abnormal	Normal
Cargo vessels	138242	3362	134890
Passenger vessels	43879	2914	40965
Tugs and Pilot vessels	50372	2306	48066

data, we pick the anomalous trajectories after applying LSTM prediction learning and LSTM wild bootstrapping methods at fixed anomaly level $(1-\alpha) = 0.95$, and use those as reference for point-wise based models performance investigation.

Selection of a Neighbourhood Function. The modified SOM (description can be found in subsection 3.1, 43p.) network was trained, using different neighbourhood functions in order to establish which has the best impact on the classification results. The initial experiments have been performed using the Klaipeda Cargo Vessels data set with the following learning parameters for the SOM network training: a shape of the grid is square and grid dimension is 20x20. The experimental results, presented in Table 21, show that the best classification accuracy is achieved using the Mexican hat neighbourhood function (marked in bold in Table 21). The results were compared with the classification accuracy obtained by other methods: a combination of SOM and Gaussian mixture models, introduced in [56], and classification carried out by experts. To ensure robustness of the results, additional experiments were carried

Table 21: Influence of the neighbourhood function on the classification accuracy when the SOM grid dimension is 20x20 (Klaipeda data set)

	Neighbour-hood function	TP	FP	TN	FN	Precision	Sensitivity
Expert		1681	0	27167	0	1	1
SOM_GMM	Gaussian	1489	81	27086	192	0.948	0.886
SOM_Pheromone	Gaussian	1477	68	27099	204	0.956	0.879
	Triangular	1241	122	27045	440	0.911	0.738
	Bubble	1454	68	27099	227	0.955	0.865
	Cut Gaussian	1479	65	27102	202	0.958	0.880
	Mexican hat	**1509**	**51**	**27116**	**172**	**0.967**	**0.898**

Table 22: Influence of the neighbourhood function on the classification accuracy when the SOM grid dimension is 25x25 (Klaipeda data set)

	Neighbour-hood function	TP	FP	TN	FN	Precision	Sensitivity
Expert		1681	0	27167	0	1	1
SOM_GMM	Gaussian	1495	80	27087	186	0.949	0.889
SOM_Pheromone	Gaussian	1491	59	27108	190	0.962	0.887
	Triangular	1288	117	25948	1495	0.917	0.463
	Bubble	1455	63	27104	226	0.955	0.865
	Cut Gaussian	1498	55	27112	183	0.958	0.866
	Mexican hat	**1512**	**50**	**27117**	**169**	**0.968**	**0.899**

out with different grid dimensions. An example of the influence of the neighbourhood function on the classification accuracy with SOM grid dimension 25x25 is presented in Table 22. In all further experiments, the Mexican hat neighboring function will be used as a reference.

Dependence of the classification accuracy on the SOM grid dimension. The comparison of the classification results obtained by the proposed algorithm SOM_Pheromone and the SOM_GMM algorithm is presented in Table 23. The experiments have been performed using the Cargo vessels data set. All experiments have been performed under the same conditions with the same parameters by increasing the SOM network grid dimension from 10x10 to 40x40 in steps of 5. By comparing the obtained results (shown in Table 23), it can be concluded that using the SOM_Pheromone algorithm for the Cargo vessels data set from Klaipeda region, the classification accuracy is better than that of SOM_GMM (the best results are marked in bold for each grid dimension). Another conclusion from the obtained results is that the optimal size of the SOM grid for the SOM_Pheromone and SOM_GMM is 25x25.

Table 23: Influence of the SOM grid dimension on the classification accuracy of SOM_Pheromone and SOM_GMM algorithms

Grid dimension	SOM_Pheromone		SOM_GMM	
	Precision	Sensitivity	Precision	Sensitivity
10x10	**0.919**	0.773	0.867	**0.780**
15x15	**0.933**	0.814	0.921	**0.834**
20x20	**0.967**	**0.898**	0.948	0.886
25x25	**0.968**	**0.899**	0.949	0.889
30x30	**0.961**	**0.897**	0.948	0.888
35x35	**0.948**	**0.893**	0.932	0.877
40x40	0.918	**0.886**	**0.919**	0.875

Table 24: Classification results of the Passenger vessels data set (normal states: 8193, abnormal states: 1457)

Method	TP	FP	TN	FN	Precision	Sensitivity
SOM_GMM	1314	17	**8176**	143	0.987	0.902
SOM_Pheromone	**1328**	**18**	8175	**123**	0.987	**0.911**

The experiment was repeated with the Passenger vessels data set (Klaipeda data set) and the Tugs and Pilot vessels data set. The classification results are presented in Tables 24 and 25. The learning parameters for the SOM network training are the same as in the previous experiment, the grid size of SOM is 25x25. The experimental results, which are presented in Table 24, show that, using the SOM_Pheromone algorithm, the best classification accuracy (marked in bold in Tables 24 and 25) is achieved.

Table 25: Classification results of the Tugs and Pilot vessels data set (normal states: 12298, abnormal states: 1153)

Method	TP	FP	TN	FN	Precision	Sensitivity
SOM_GMM	971	9	12289	182	0.991	0.842
SOM_Pheromone	**978**	9	12289	**175**	0.991	**0.848**

Retraining strategies

Due to the amount of time required by semi-supervised training step, the performance of a method on different retraining strategies, as described in subsection 3.3 on 49 - 52 pages, were investigated. Initially the data was split to three batches (T1, T2, T3; see Figure 11, p. 51) ordered by data gathering timestamp. Overall, three strategies were developed: Strategy I - the SOM network is retrained every time from beginning as data arrives; Strategy II - the SOM network is retrained only with newly arrived data. Strategy III - the SOM network is retrained with mixture of newly arrived and historical data.

Strategy I. For the SOM network training and validation, we used T1, T2 and T3 data batches. The learning rate parameter was set to 0.5. Then, after the network was trained and validated with the T1 data batch, the new data were fed to the network as follows: the T1 and T2 batch data were merged and the algorithm was trained from the initial random state using all items from T1 and T2. The same scheme was applied to the T3 data batch.

In order to achieve the best network performance, the learning rate parameter can be adjusted. Initial research led us to divide the learning rate parameter search into these intervals and step sizes: in the interval $[0.005; 0.04]$, step was set to 0.005; in the interval $[0.04; 0.1]$, step size was increased to 0.01; and, in the interval $[0.1; 0.5]$, step size was set to 0.1 (see Table 26). In this way, the training experiment of Strategy I was repeated while every learning parameter value was tested to achieve the best algorithm performance. After the model was trained, it was tested with the test data set, which allowed to evaluate the general model error. The best-obtained model characteristics with model test data set are presented in Table 26 (the row in bold).

The statistics of the best Strategy I model using test data for general model error estimation and test data for model error estimation is presented in Table 27. The time needed for the algorithm retraining was 40,769 s.

Strategy II. The initial algorithm was trained 10 times with the T1

Table 26: Selection of learning rate

Learning Rate	TP	FP	TN	FN	Precision	Sensitivity
0.005	924	519	26648	757	0.6403	0.5497
0.010	943	505	26662	738	0.6512	0.5610
0.015	957	498	26669	724	0.6577	0.5693
0.020	963	487	26680	718	0.6641	0.5729
0.025	968	478	26689	713	0.6694	0.5758
0.030	976	471	26696	705	0.6745	0.5806
0.035	986	468	26699	695	0.6781	0.5866
0.040	998	461	26706	683	0.6840	0.5937
0.050	1025	445	26722	656	0.6973	0.6098
0.060	1066	413	26754	615	0.7208	0.6341
0.070	1109	394	26773	572	0.7379	0.6597
0.100	1197	303	26864	484	0.7980	0.7121
0.200	1431	135	27032	250	0.9138	0.8513
0.300	1486	81	27086	195	0.9483	0.8840
0.400	1500	55	27112	181	0.9646	0.8923
0.500	**1510**	**52**	**27115**	**171**	**0.9667**	**0.8983**
0.600	1507	54	27113	174	0.9654	0.8965
0.700	1502	59	27108	179	0.9622	0.8935

Table 27: Training Strategy I performance at learning rate 0.5

Stage	TP	FP	TN	FN	Precision	Sensitivity
Testing (model error)	1510	52	27115	171	0.9667	0.8983
Testing (general error)	1868	69	33890	233	0.9644	0.8891

batch data. During each training, the weights of the SOM network were generated randomly, and the best performing network was selected while keeping a fixed learning rate parameter at the value of 0.5. The performance of the investigated network on repetitive Strategy II (using only T1 data set) model evaluation and testing is presented in Table 28. The row in bold indicates the best network obtained. Quite small deviations in precision and sensitivity rates show the stability of the network. Then, parameters of the best-obtained network were used as initial weights for the network to be trained with T2 batch data. Finally, imitating the new data portion arrival, the best model obtained with T2 batch data was retrained with the T3 batch data. The results of the additional experiment show that the best performance network was obtained with learning rate of 0.025. The statistics (model test error and general model error evaluation) of the best model data are presented in Table 29. The time needed for model training was 18, 229 s. Strategy III. The scheme of the model training validation and testing was similar to that described in Strategy II, except for the following two aspects. First, four data batches (Tm2–Tm5) were produced from from T2 and T3 batches. Each

Table 28: Strategy II performance on model test data

No.	TP	FP	TN	FN	Precision	Sensitivity
1	1364	241	26926	317	0.8498	0.8114
2	1329	280	26887	352	0.8260	0.7906
3	1359	252	26915	322	0.8436	0.8084
4	1364	274	26893	317	0.8327	0.8114
5	1356	253	26914	325	0.8428	0.8067
6	1335	253	26914	346	0.8407	0.7942
7	1314	251	26916	367	0.8396	0.7817
8	1332	258	26909	349	0.8377	0.7924
9	**1367**	**237**	**26930**	**314**	**0.8522**	**0.8132**
10	1338	240	26927	343	0.8497	0.7960
				max	0.8522	0.8132
				min	0.8260	0.7817
				average	0.8413	0.8011
				stdev	0.0079	0.0115

Table 29: Retraining Strategy II performance at learning rate 0.025

Stage	TP	FP	TN	FN	Precision	Sensitivity
Testing (model error)	1500	98	27069	181	0.9387	0.8923
Testing (general error)	1836	122	33837	265	0.9377	0.8739

new batch contains one quarter of both T2 and T3 data (see Table 30). Second, as previously described, after every model training and validation, the parameters of the best-obtained model were used for every next Tm2–Tm5 batch training, except for the model training data aggregation. In every retraining cycle, the model error was estimated the same way as described in Strategies I and II. Half of the items from Tm2–Tm5 data batches consisted of items from T2 and T3, as shown in Table 30 (Tm2–Tm5), while another part of the data was selected proportionally, with respect to those data points attached to the previous best model SOM winning neurons. This approach guaranteed that the knowledge of frequently passed sea regions was incorporated into the next model

Table 30: Partitioning of data set (Strategy III)

Data Batches	% of Train and Validation Data	New Data Items	All Data Items
T1	60%	69235	69235
Tm2	10%	11539	23078
Tm3	10%	11539	23078
Tm4	10%	11539	23078
Tm5	10%	11539	23078

training because ships do not change their sea routes often. Experiments show that the best model was obtained with the learning rate of 0.03.

The statistics for the Strategy III best model were obtained using test data for general model error estimation, and the results are presented in Table 31.

Table 31: Retraining Strategy III performance at learning rate of 0.003

Stage	TP	FP	TN	FN	Precision	Sensitivity
Testing (model error)	1527	73	27094	154	0.9544	0.9084
Testing (general error)	1866	91	33868	235	0.9535	0.8881

The time needed for the algorithm retraining was $27,854$ s. The summary of relative time required for the training Strategies I–III is presented in Table 32. The same data batching Strategies I–III that were described above were tested on the Passenger data set as well. The results are presented in Table 33. Results in Tables 32 and 33 show that, by applying different SOM model retraining strategies and keeping the same data batch sizes, it is possible to substantially decrease the training time for detection of maritime traffic abnormal movements while retraining the model precision and sensitivity at very high values. The obtained results show that the SOM network could be retrained in half the time while keeping precision and sensitivity at almost the same high values.

Table 32: Retraining Strategies I–III performance on Cargo data set

Strategy	Precision	Sensitivity	Relative Time
Strategy I	0.9644	0.8891	1
Strategy II	0.9377	0.8739	0.4471
Strategy III	0.9535	0.8881	0.6832

Table 33: Retraining Strategies I–III performance on Passenger data set

Strategy	Precision	Sensitivity	Relative Time
Strategy I	0.9795	0.8897	1
Strategy II	0.9802	0.8870	0.4478
Strategy III	0.9817	0.8888	0.6817

SOM-Pheromone and SOM-GMM methods and "Fehmarnbelt" data set. The experimental setup is applied to the "Fehmarnbelt" data set in order to study performance of SOM methods on significantly larger and intense marine traffic areas. The data was prepared for training and testing. The data summary is shown in Table 14 on page 41. Further in this subsection the quantities of data will be shown as vectors, where single sequence contains $n' + \widetilde{n} = 100$ navigational vectors.

As mentioned earlier, Klaipeda's data set is Expert annotated, but "Fehmarnbelt" is not. For both SOM based methods it is essential to have the labeled data for fine-tuning. Based on those labels, the SOM_Pheromone's β_{PPV}, β_{TPR} and SOM_GMM's $P(H = normal)$ parameters are fine-tuned to maximize sensitivity and precision of anomaly detection. The "Fehmarnbelt" data set is enormous and unlabeled. To label such a tremendous amount of data is unfeasible because that requires many work hours of costly expert work and makes semi-supervised algorithms too expensive. To compare performance of SOM based methods with LSTM based methods, we need anomaly labels for vessel trajectories/navigational vectors. In order to collect such anomaly labels, the LSTM prediction learning and LSTM wild bootstrapping models were trained for anomaly level of $(1 - \alpha) = 0.95$. These obtained anomalous vectors were used as a reference for SOM-based methods for fine-tuning model parameters. Moreover, the grid size, neighborhood function, and learning rate are selected the same way as depicted in the experiment with Klaipeda's data set. Both SOM based methods were trained on all distinct vessel types separately while estimating model parameters. The calibration, fine-tuning, and validation are performed independently on a part of data set, that was produced by LSTM prediction learning and LSTM wild bootstrapping independently and labeled as anomaly..

In order to retain visual intuitiveness, the SOM_pheromone and SOM_GMM networks are displayed only on the geographical plain by using only Latitude and Longitude features from SOM neuron weights (code-books). Such visualization hides other vector/codebook features, and because of that, it cannot be used to evaluate the model. This visualization is meant to give VTS operator a general understanding of normal traffic distribution in the geographical area. For this intuitive visualization Figures 25 and 26 are presented.

Figure 25 shows SOM_pheromone longitude and latitude parts of neuron weights/codebook for grid size 60 × 60, that was trained using traffic of fishing type vessels at "Fehmarnbelt" region. The figure shows longitude and latitude of navigational vectors ("colored dot") and SOM neurons ("×") on two dimensional geographical plain. It can be observed that in areas of higher traffic intensity, where navigational vectors are more dense, the SOM neurons create clusters that cover a smaller geographical area. It gives smaller granularity for further model creation for anomaly detection. Experiments have shown (see Table 23) that the size of SOM grid influences precision of both SOM_pheromone and SOM_GMM models. For "Fehmarnbelt" data set analysis, each vessel type was investigated with respect to obtain the best SOM grid size and

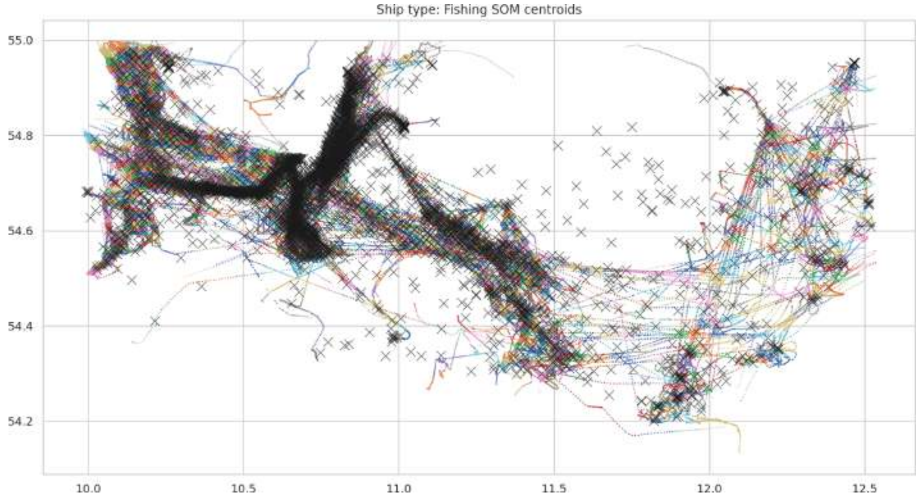


Figure 25: SOM neurons grid 60×60 visualization of Fishing type vessels traffic

neighbourhood function (see Table 34) with respect to fixed PICP by LSTM methods described above. The visualisations of SOM grids for other vessel types can be found in Appendix G.

Figure 26 shows negative likelihood ($-log(P(H = normal))$) of Fishing type vessel traffic formed by trained SOM_GMM model. Negative likelihood is chosen to visualise contour plot of likelihood spread across geographical region more clearly. The blue color shows navigational vector projection on geographical plain of vessel traffic. The contour lines depict negative likelihood of SOM_GMM model. The darker color means higher $-log(P(H = normal))$ likelihood, and brighter colour means lower likelihood. It can be observed that darker color falls on the denser part of vessel traffic and by adjusting likelihood threshold $P(H = normal)$ we could fine tune sensitivity of anomaly detection.

Table 34 shows final results of SOM based method experiments, which were performed using "Fehmarnbelt" data set grouped by specific vessel type. The above experiment data follows the same workflow as that performed on Klaipeda's data set: precision, sensitivity and PICP dependency were investigated using selection of neighbourhood functions and grid sizes. In summary, the suggested (see Subsection 3.3, p. 49) retraining strategies also showed same patterns as those observed by analysing Klaipeda's data set. Strategy III took only 0.671 fraction of computational time required by Strategy I with precision drop by average of 0.007 and 0.009 drop in sensitivity. Learning rate parameter bound for Strategy III is between 0.03 and 0.04. Thus the results of "Fehmarnbelt" data set show that the retraining strategies can be applied to minimize

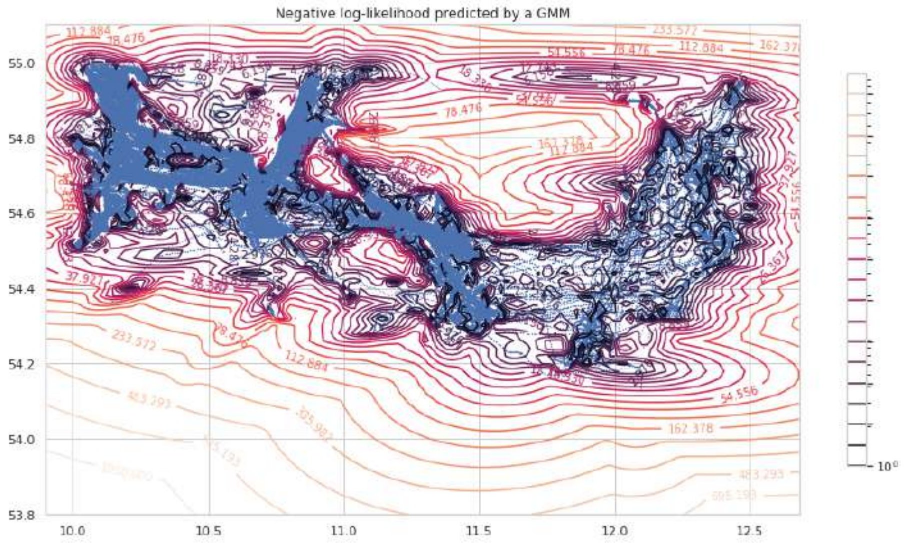


Figure 26: SOM_GMM's negative log likelihood prediction of fishing type vessels traffic

training time while keeping the sensitivity and precision at feasible levels for SOM_pheromone and SOM_GMM point-wise algorithm modifications.

Additionally, it was observed that larger data sets such as Cargo, Passenger, and Tanker require significantly larger grid sizes (70 × 70) in comparison with the smaller data sets such as Diving, HSC, Towing, etc. vessels (35×35) (see Table 34) SOM. Also, the Cut Gaussian neighbourhood function performs better on larger data sets, yet Mexican hat neighbourhood function works better on the smaller ones. Moreover, SOM_GMM requires smaller grid sizes than the SOM_Pheromone (see the SOM_pheromone and SOM_GMM Grid column in Table 34). Also, we can see that SOM_GMM is less sensitive for grid size variance and is more precise on larger data sets. However, SOM_pheromone method performs better on smaller data sets (eg. Anti-pollution, Diving). Furthermore, the SOM_Pheromone and SOM_GMM fine tuned models were validated using LSTM wild bootstrapping model output. The results were similar when the models were tested using reference labels from both LSTM methods, except for very small data sets for which statistical LSTM wild bootstrap method failed to learn prediction region (results are shown as "-").

Also, it can be observed that both SOM based methods have low sensitivity values. For small data sets (e.g. Spare_1, HSC, Diving) the sensitivity is better, but in general it is low. This fact shows that such

Table 34: SOM_pheromone and SOM_GMM experiment results using "Fehmarnbelt" data set

	Vectors $,\times 10^2$		LSTM prediction region								LSTM wild bootstrap				
				SOM-pheromone				SOM-GMM				SOM-pheromone		SOM-GMM	
Vessel type	train	test	PICP	Grid	NF	PPV	TPR	Grid	PPV	TPR	PICP	PPV	TPR	PPV	TPR
Anti-pollution	1094	365	0.941	40x40	MH	0.901	0.774	35x35	0.886	0.740	0.491	0.803	0.747	0.897	0.721
Cargo	75625	25209	0.960	70x70	CG	0.675	0.667	60x60	0.856	0.634	0.975	0.719	0.680	0.824	0.608
Diving	103	35	0.835	35x35	MH	0.914	0.869	30x30	0.899	0.896	0	-	-	-	-
Dredging	4584	1528	0.951	70x70	CG	0.667	0.640	55x55	0.844	0.601	0.892	0.720	0.636	0.865	0.692
Fishing	15748	5250	0.944	60x60	CG	0.626	0.656	60x60	0.849	0.631	0.897	0.713	0.652	0.849	0.691
HSC	32	11	0.846	35x35	MH	0.914	0.947	30x30	0.904	0.936	0.009	-	-	-	-
Law-enforcement	4467	1489	0.955	70x70	CG	0.638	0.555	55x55	0.857	0.600	0.816	0.722	0.576	0.841	0.679
Military	4743	1581	0.953	70x70	CG	0.667	0.700	60x60	0.842	0.586	0.824	0.716	0.667	0.843	0.666
Passenger	47988	15996	0.959	70x70	CG	0.671	0.617	60x60	0.806	0.550	0.964	0.679	0.557	0.845	0.556
Pilot	6352	2118	0.959	70x70	CG	0.658	0.683	40x40	0.859	0.648	0.829	0.723	0.601	0.832	0.553
Pleasure	1965	655	0.943	60x60	CG	0.660	0.601	50x50	0.846	0.635	0.868	0.743	0.604	0.851	0.576
Port-tender	272	91	0.487	35x35	MH	0.898	0.761	35x35	0.898	0.748	0	-	-	-	-
Reserved	436	146	0.947	30x30	MH	0.908	0.880	30x30	0.903	0.893	0	-	-	-	-
SAR	3704	1235	0.959	60x60	CG	0.612	0.605	55x55	0.848	0.619	0.962	0.736	0.681	0.851	0.691
Sailing	6692	2231	0.952	70x70	CG	0.662	0.602	60x60	0.855	0.629	0.981	0.748	0.693	0.856	0.618
Spare_1	15	6	0.903	30x30	MH	0.911	0.879	30x30	0.909	0.862	0.074	-	-	-	-
Tanker	22577	7526	0.946	70x70	CG	0.603	0.573	55x55	0.859	0.670	0.969	0.752	0.685	0.830	0.644
Towing	522	175	0.944	30x30	MH	0.890	0.788	30x30	0.901	0.844	0	-	-	-	-
Towing-long-wide	367	123	0.925	30x30	MH	0.913	0.885	30x30	0.907	0.892	0	-	-	-	-
Tug	9421	3141	0.955	60x60	CG	0.641	0.654	50x50	0.839	0.644	0.945	0.725	0.633	0.864	0.655
WIG	40	14	0.726	30x30	MH	0.916	0.937	30x30	0.911	0.940	0	-	-	-	-
Mean:						0.759	0.727		0.871	0.724		0.731	0.647	0.850	0.642

Abbreviations in table: NF - neighbourhood function; PPV - Precision; TPR - Sensitivity; MH - Mexican hat; CG - Cut Gaussian;

models produce a high number of false negative cases. The higher number of false negatives, the harder these methods detect traffic anomalies that can be detected by LSTM methods. This assumption more thoroughly inspected in the subsection 5.4 *"Comparison of Anomalous Traffic Trajectories"*.

5.4 Comparison of Anomalous Traffic Trajectories

In this subsection, the investigation of the number of false negatives is performed. As a tool, the Spatio-temporal clustering method is applied. Cuturi and Blondel [93] propose soft-DTW k-means algorithm to cluster time series data. In order to apply the idea in the research, the multivariate version of the proposed algorithm is used. The sequences of predicted trajectory latitude and longitude point coordinates from X (see Equation (6) on 35 p.) were used as input data for algorithm. Figure 27 summarizes the result of clustered anomalous trajectories. The

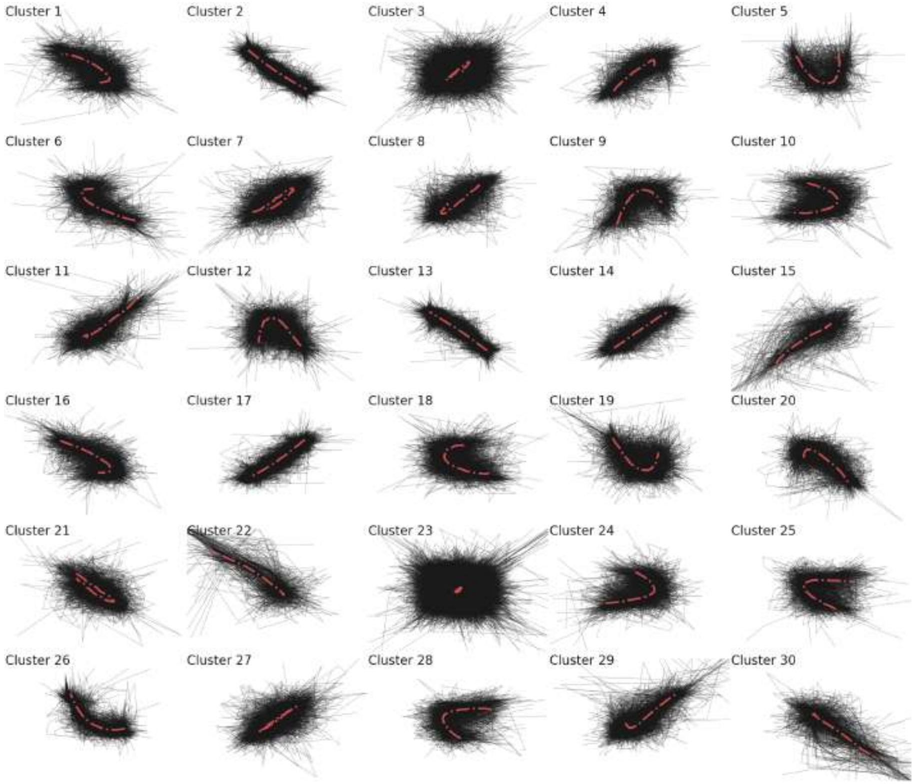


Figure 27: Clustered anomalous vessel traffic trajectories of Fishing vessel type

figure represents sets of vessel navigational vector sequences from the anomalous false-negative vessel trajectories. The trajectories assigned to the particular cluster are colored in black, and the barycenter of the cluster is colored in red. The number of clusters was chosen using an elbow method [94].

Navigational sequences are pre-processed using scaling to zero mean and unit variance. The assumption is that the range of a given sequence is uninformative, and one only wants to compare trajectory shapes in an amplitude-invariant manner. As the sequences are multivariate, the scaling re-scales all modalities so that there will not be a single modality responsible for a large part of the variance. This approach means that scale barycenters of the sequence are scaled independently, and there is no such thing as an overall data range [93, 95]. Thus the approach clusters the data only by the trajectory shape but not location.

Figure 27 shows 30 trajectory clusters of fishing vessel type. One may observe that the red lines depict rather different trajectory shapes. In addition to that, clusters with numbers 2, 13, 14, and 17 share the

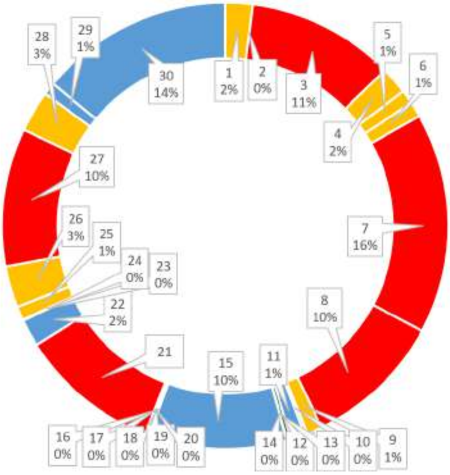


Figure 28: Analysis of Fishing vessel anomalous trajectory groups of SOM_GMM false negatives

same shape properties, except for different vessel movement directions. This remark extends to other clusters as well. Thus, by taking into account, the marine vessel domain knowledge and shape rotations obtained 30 clusters forms in 5 distinct cluster groups that represent anomalous trajectory shapes:

- Straight trajectory line shape cluster - 2, 13, 14, 17 (Green).
- Stopping trajectory line shape cluster - 11, 15, 22, 29, 30 (Blue).
- Soft manoeuvre trajectory shape cluster - 1, 4, 5, 6, 9, 10, 12, 16, 18, 19, 20, 24, 25, 26, 28 (Yellow).
- Sharp manoeuvre trajectory shape cluster - 3, 7, 8, 21, 27 (Red).
- Drift trajectory shape cluster 23 (White).

Figure 28 shows the distribution of false-negative trajectory shapes among mentioned cluster groups. One can see that Sharp maneuver trajectory shape clusters (marked red) form the majority of shapes (57.00%). Second in size is the Stopping trajectory line shape cluster (marked blue) with the amount of 27.9% of trajectories in the set of false negatives, while Soft maneuver trajectory clusters constitute 15,07%. Finally, the Straight and Drift trajectory clusters constitute 0.02% and 0.005% of false-negative shapes respectively. A similar clustering trend is observed in the larger sets of vessel types such as Anti-pollution, Cargo, Dredging, Fishing, Law_enforcement, Military, Passenger, Pilot, Pleasure, SAR, Sailing, Tanker, Tug, and in smaller vessel data sets grouped by vessel type. The analysis of the false-negative trajectory line shapes concludes that obtained false-negatives dominate the line shapes of Soft maneuver and Straight trajectories, and constitute on average 67% and 21% of the whole sample set, respectively.

5.5 Conclusions of the Section

During the experimental investigation, it was observed that all models behave differently on two data set groups based on their size. The first group is less than 138242 navigational vectors. This group contains the following data sets: Klaipeda Cargo, Passenger, Tugs, and Pilot; Fehmarnbelt Anti-pollution, Diving, HSC, Port_tender, Reserved, Spare_1, Towing, Towing_long_wide, and WIG. The second group exceeds 138242 and contains the following data sets: Fehmarnbelt Cargo, Dredging, Fishing, Law_enforcement, Military, Pilot, Pleasure, SAR, Tanker, Sailing, Tanker, and Tug.

Semi-supervised SOM methods The SOM_pheromone and SOM_GMM methods were tested using Klaipeda and "Fehmarnbelt" sea regions vessel traffic AIS data sets. When data for the Klaipeda region are used, the proposed SOM_Pheromone modification outperforms the SOM_GMM algorithm, but on "Fehmarnbelt", the SOM_pheromone outperformed only on smaller Anti-pollution, Diving, HSC, Spare_1, Towing-long-wide, WIG vessel type data sets. On other larger vessel type data sets, the SOM_GMM shows better results of classification precision.

The SOM_pheromone algorithm performs better than SOM_GMM on smaller data sets. For this data set group, the recommended SOM_pheromone parameter is the Mexican Hat neighbourhood function, grid size from 35 × 35 to 40 × 40. On smaller data sets, the method reaches a precision of 0.982, with a sensitivity of 0.889 (Klaipeda passenger data set). On the larger data set group, SOM_GMM performs better than SOM_pheromone. The recommended parameters are: Cut Gaussian neighbourhood function, grid size from 55 × 55 to 60 × 60. The best-obtained precision is 0.859, and sensitivity is 0.648. For larger data sets, in order to obtain the best result, SOM_GMM requires a smaller grid size (60 × 60) compared with SOM_pheromone (70 × 70).

Both SOM methods have shown better performance on the smaller data set group when a Mexican hat neighbourhood is used. However, for the larger group, the Cut Gaussian function shows better performance results.

During testing of differed sizes of SOM grid, it was observed that performance depends on the size of the data set and algorithm. For the Klaipeda data set, the best precision was shown by 25×25 grid. For best precision in the larger "Fehmarnbelt" data set group, SOM_pheromome requires 60 × 60 to 70 × 70 grid size and SOM_GMM requires 45 × 45 to 60 × 60. For the smaller data set group, SOM_pheromome requires 30 × 30 to 35 × 35, and SOM_GMM requires 30 × 30 to 35 × 35. For the larger data set group, SOM_GMM requires a smaller SOM grid in comparison with SOM_pheromone method.

SOM retraining strategies By applying different SOM model retraining strategies, while keeping the exact data batch sizes, it is possible to substantially decrease the time for detection of maritime traffic abnormal movement while the model is retrained for precision and sensitivity at very high values. The results obtained show that the SOM network could be retrained in half the time while keeping precision and sensitivity at almost the same high values.

If the model is trained from the initial random weights of the SOM network, the best performance is observed; however, the training time is the longest. Model precision reaches 0.979, and sensitivity is 0.889 at a learning rate of 0.5.

If the model is trained on top of the pretrained model weights, the precision and sensitivity drop slightly, but the training time decreases by half at a learning rate of 0.025.

Let us suppose that the model is trained on top of the pretrained model weights, and the newly arrived data batch is proportionally mixed with those winning neurons. In that case, the training time can be decreased by one-third while keeping almost the same results as depicted previously at a learning rate of 0.03.

The suggested retraining Strategy III took only 67.1% of computational time required by Strategy I method with the precision drop to the range from 0.007 to 0.009. The learning rate parameter for the proposed strategy is between 0.03 and 0.04.

Unsupervised LSTM methods LSTM multi-stacked multivariate auto-encoder was used to predict vessel trajectories in an unsupervised approach. For smaller data set groups, the LSTM prediction region learning method and the LSTM bootstrap have MAE average error of 11.14 and 10.71 km, respectively, for the group. For the larger group, the average error is 2.73 and 2.07 km, respectively. It shows that the LSTM method is more accurate for the larger data set group than for the smaller data set group with a difference of 8.41 and 8.64 km.

The Prediction Region Coverage Probability (PICP) value was controlled in LSTM prediction learning method by adjusting λ value. It was observed that when λ value is increased linearly, the PICP and PINAW values increase logarithmically. Each vessel type data set has a particular

λ value that corresponds to PICP value.

By testing the PICP value on validation and test data set, it was observed that models of smaller data set groups show significantly lower values when the goal is $PICP = 95\%$. (Diving - 83.5%, HSC - 84.6%, Port_tender - 48.7%, Spare_1 - 90.3%, WIG - 87.3%). PICP values of larger data set groups are closer to the desired values (94.1%, 94.4%, 95.1%, etc.). It shows that LSTM prediction region learning method is less accurate on smaller data sets than on the larger data set group. Additionally, after performing 10 LSTM networks learning with the same λ value, but different initial random weights, it was observed that the learned set of PICP values on the smaller data set group have significantly larger variances compared to the larger data set group.

The prediction region results of both LSTM methods show that for the larger data set group, the PICP values are close to predefined $100(1-\alpha) = 95\%$ with values in the range 94.1% to 97.5%. On the smaller data set group, the LSTM prediction learning method can still learn with narrower prediction regions from 48.7% to 84.6%. The LSTM wild bootstrapping method was unable to learn prediction regions for the smaller data set group with the PICP value of 0. Both LSTM methods for unsupervised estimation of prediction regions can be used for abnormal marine traffic detection when training data sets in the larger group. It is recommended to use LSTM prediction region learning method with narrower prediction regions for the smaller data sets.

Trajectory line shapes After fine-tuning both SOM methods with anomalous traffic labels, which were detected by LSTM methods, it was observed that SOM methods have low classification sensitivity values and, accordingly, large false negative values, especially in the larger data set group. Anomalous vessel traffic trajectories were clustered into 30 clusters representing five general type trajectory shapes: straight line, stopping line, soft maneuver, sharp maneuver, and drift maneuver. False negatives of SOM methods constitute 57.0% of sharp maneuver trajectories, 27.9% of stopping trajectory line shapes, 15.07% of soft maneuver trajectory line shapes, 0.02% of straight trajectory line shapes, and 0.005% of drift trajectory line shapes. This observation shows that SOM methods detect trajectory anomalies of this type significantly worse than LSTM methods.

GENERAL CONCLUSIONS

1. The accuracy of SOM_pheromone and SOM_GMM methods depend on amount of data. It was observed that SOM_pheromone method detects anomalous traffic more accurately than SOM_GMM method does in data up to around 140,000 navigational vectors. Experiments show that the recommended SOM_pheromone parameters are: Mexican Hat neighbourhood function and grid size from 35×35 to 40×40. On smaller data sets, an average precision is 0.911 and sensitivity is 0.801 versus SOM_GMM's precision of 0.886 and sensitivity of 0.740. On larger data sets with more than approximately 140,000 navigational vectors, the SOM_GMM outperforms SOM_pheromone. The SOM_GMM's average precision is 0.859 and sensitivity is 0.648 in comparison with SOM_pheromone's average precision of 0.675 and sensitivity of 0.640. The recommended parameters are Cut Gaussian neighbourhood function and grid size from 55×55 to 60×60.

2. It is possible to substantially decrease model training time to detect abnormal maritime traffic movement when the model precision and sensitivity retain high values. Proposed SOM retraining strategy is applied to achieve this goal, where neurons' previous trained weights are used as a starting position for retraining the network with newly gathered data subset mixing it with historical data and adjusting the learning rate. The obtained results show that the SOM_pheromone and SOM_GMM networks could be retrained in half the time while keeping precision and sensitivity at almost the same high values. The suggested retraining strategy took only 67.1% of computational time required by the classical method with the precision drop to the range from 0.007 to 0.009. The learning rate parameter for the proposed strategy is between 0.03 and 0.04.

3. The prediction region results of both LSTM methods show that for larger data sets with more than approximately 140,000 navigational vectors, the Prediction Region Coverage Probability (PICP) value is close to the predefined $100(1-\alpha) = 95\%$ value, when values are in range from 94.1% to 97.5%. On smaller data sets with less than approximately 140,000 navigational vectors, the LSTM prediction learning method was still able to learn with narrower prediction regions from 48.7% to 84.6%. The LSTM wild bootstrapping method was unable to learn prediction regions for smaller data sets, which is indicated by the PICP value of 0. Both LSTM algorithms for unsupervised estimation of prediction regions can be used for the detection of abnormal marine traffic when train-

ing data sets are larger than approximately 140,000 navigational vectors. For smaller data sets, it is recommended to use LSTM prediction region learning method with narrower prediction regions.

4. In order to solve the issue of missing vessel type data, a multistacked multivariate LSTM classifier was developed. The proposed model performs well, the average precision is 0.96079, the average sensitivity is 0.96060, and f1-score is 0.96056. Classification metrics show good generalization properties that allow to perform imputation and gain classes for the 4.28% percent with missing feature value (from a total of 4234160 navigational vectors) in the "Fehmarnbelt" data set.

5. The test results of SOM methods, where LSTM output was taken as class reference, show low values of sensitivity (from 0.555 to 0.700) due to high values of false negatives. The analysis of these false negative results allows to conclude that sharp manoeuvre trajectory and stopping trajectory line shapes dominate in obtained false negatives set and constitute an average of 57.0% and 27.9% in the larger data set group.

References

[1] The European Commission. *Maritime Transport Statistics-Short Sea Shipping of Goods.* `https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Maritime_transport_of_goods_-_quarterly_data`. (accessed: 21.08.2020).

[2] Z. Wan et al. "Four routes to better maritime governance". In: *Nature* 540 (2016), pp. 127–29.

[3] P. Fu et al. "Finding Abnormal Vessel Trajectories Using Feature Learning". In: *Nature* 5 (2017), pp. 7898–7909.

[4] J. Will, L. Peel, and C. Claxton. "In Proceedings of the IMA Maths in Defence Conference, Swindon, UK". In: *the IMA Maths in Defence Conference* 20 October (2011).

[5] Zhixiang He, Chi-Yin Chow, and Jia-Dong Zhang. "STNN: A Spatio-Temporal Neural Network for Traffic Predictions". In: *IEEE Transactions on Intelligent Transportation Systems* 1.1 (2020), pp. 1–10. ISSN: 1524-9050. DOI: `10.1109/tits.2020.3006227`.

[6] D. Li et al. "Smoothed LSTM-AE: A spatio-temporal deep model for multiple time-series missing imputation". In: *Neurocomputing* 411 (2020), pp. 351–363. ISSN: 18728286. DOI: `10.1016/j.neucom.2020.05.033`. URL: `https://doi.org/10.1016/j.neucom.2020.05.033`.

[7] S. Tian et al. "Spatio-Temporal position prediction model for mobile users based on LSTM". In: *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS* 2019-Decem (2019), pp. 967–970. ISSN: 15219097. DOI: `10.1109/ICPADS47876.2019.00146`.

[8] Centre Of Excellence For Operations In Confined And Shallow Waters et al. *Maritime Situation Awareness.* 2015. URL: `https://www.coecsw.org/fileadmin/content_uploads/projects/20150423_MSA_Study_Paper_-_Final.pdf`.

[9] E Martineau and J Roy. *Maritime Anomaly Detection: Domain Introduction and Review of Selected Literature.* 2011.

[10] Filipe Dias et al. "Maritime Situational Awareness, the singular approach of a dual-use Navy". In: *Scientific Bulletin of Naval Academy* XXI (July 2018), pp. 203–215. DOI: `10.21279/1454-864X-18-I1-033`.

[11] A. Sidibé and G. Shu. "Study of automatic anomalous behaviour detection techniques for maritime vessels". In: *J. Navig* 70 (2017), pp. 847–858.

[12] V. Fernandez Arguedas, G. Pallotta, and M. Vespe. "Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring". In: *IEEE Transactions on Intelligent Transportation Systems* 19.3 (2018), pp. 722–732. ISSN: 15249050. DOI: 10.1109/TITS.2017.2699635.

[13] Virginia Fernandez Arguedas, Fabio Mazzarella, and Michele Vespe. "Spatio-temporal data mining for maritime situational awareness". In: *MTS/IEEE OCEANS 2015 - Genova, Italy*. May 2015, pp. 1–8. DOI: 10.1109/OCEANS-Genova.2015.7271544.

[14] M.J. Riveiro. "Visual analytics for maritime anomaly detection". PhD thesis. Orebro universitet, 2011.

[15] Centre Of Excellence For Operations In Confined And Shallow Waters. *The Role and Relevance of the Maritime Domain in an Urban-Centric Operational Environment*. 2017. URL: https://www.coecsw.org/fileadmin/content_uploads/projects/Role_and_Relevance_of_the_Maritime_Domain_in_an_Urban-Centric_Operational_Environment.pdf.

[16] Jan Ekman and Anders Holst. "Incremental stream clustering and anomaly detection". In: *SICS Technical Report* 1 (Jan. 2008), p. 55. ISSN: ISSN 1100-3154.

[17] N. Lu et al. "Shape-Based Vessel Trajectory Similarity Computing and Clustering: A Brief Review". In: *2020 5th IEEE International Conference on Big Data Analytics, ICBDA 2020* (2020), pp. 186–192. DOI: 10.1109/ICBDA49040.2020.9101322.

[18] Leonid Portnoy, Eleazar Eskin, and Salvatore Stolfo. "Intrusion Detection with Unlabeled Data Using Clustering". In: *In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security*. Nov. 2001.

[19] R.O. Lane et al. "Maritime anomaly detection and threat assessment". In: *In Proceedings of the FUSION 2010 : 13th International Conference on Information Fusion, Edinburgh, UK* 26–29 July 2010 (2010).

[20] International Maritime Organization. *International Convention for the Safety of Life at Sea (SOLAS)*. http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-(SOLAS),-1974.aspx. (Accessed: 29.08.2020). 1976.

[21] International Maritime Organization. *Revised Guidelines for The Onboard Operational Use Of Shipborne Automatic Identification Systems (AIS), Resolution A.1106(29).* http://www.imo.org/en/KnowledgeCentre/IndexofIMOResolutions/Assembly/Documents/A.1106(29).pdf. (Accessed: 29.08.2020). 2015.

[22] N.V. Loi et al. "Abnormal moving speed detection using combination of kernel density estimator and DBSCAN for coastal surveillance radars". In: *2020 7th International Conference on Signal Processing and Integrated Networks, SPIN 2020* (2020), pp. 143–147. DOI: 10.1109/SPIN48934.2020.9070885.

[23] B. Ristic. "Detecting Anomalies from a Multitarget Tracking Output". In: *IEEE Transactions on Aerospace and Electronic Systems* 50.1 (2014), pp. 798–803.

[24] F. Zhu. "Mining ship spatial trajectory patterns from AIS database for maritime surveillance". In: *2011 2nd IEEE International Conference on Emergency Management and Management Sciences.* 2011, pp. 772–775.

[25] S. K. Singh and F. Heymann. "Machine Learning-Assisted Anomaly Detection in Maritime Navigation using AIS Data". In: *2020 IEEE / ION Position, Location and Navigation Symposium (PLANS).* 2020, pp. 832–838.

[26] J. Venskus et al. "Integration of a Self-Organizing Map and a Virtual Pheromone for Real-Time Abnormal Movement Detection in Marine Traffic". In: *Informatica (Netherlands)* 28.2 (2017), pp. 359–374. ISSN: 08684952. DOI: 10.15388/Informatica.2017.133.

[27] J. Venskus et al. "Real-time maritime traffic anomaly detection based on sensors and history data embedding". In: *Sensors (Switzerland)* 19.17 (2019). ISSN: 14248220. DOI: 10.3390/s19173782.

[28] G. Pallotta, M. Vespe, and K. Bryan. "Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction". In: *Entropy* 15.6 (2013), pp. 2218–2245. ISSN: 10994300. DOI: 10.3390/e15062218.

[29] Andrius Daranda and Gintautas Dzemyda. "Neural network approach to predict marine traffic". In: *Baltic journal of modern computing* 4.3 (2016), pp. 483–495. ISSN: 2255-8942.

[30] Andrius Daranda and Gintautas Dzemyda. "Navigation decision support: discover of vessel traffic anomaly according to the historic marine data". In: *International journal of computers, communications and control* 15.3 (2020), pp. 1–9. ISSN: 1841-9836. DOI: https:

//doi.org/10.15837/ijccc.2020.3.3864. URL: http://www.sciencedirect.com/science/article/pii/S1053811998903913.

[31] P.A.M. Silveira, A.P. Teixeira, and C.G. Soares. "Use of AIS Data to Characterise Marine Traffic Patterns and Ship Collision Risk off the Coast of Portugal". In: *Journal of Navigation* 66.6 (2013), pp. 879–898. DOI: 10.1017/S0373463313000519.

[32] P. R. Lei. "A framework for anomaly detection in maritime trajectory behavior". In: *Knowledge and Information Systems* 47.1 (2016), pp. 189–214. ISSN: 02193116. DOI: 10.1007/s10115-015-0845-4.

[33] R. Zhen et al. "Maritime Anomaly Detection within Coastal Waters Based on Vessel Trajectory Clustering and Naïve Bayes Classifier". In: *Journal of Navigation* 70.3 (2017), pp. 648–670. ISSN: 14697785. DOI: 10.1017/S0373463316000850.

[34] K. Sheng et al. "Research on Ship Classification Based on Trajectory Features". In: *Journal of Navigation* 71.1 (2018), pp. 100–116. DOI: 10.1017/S0373463317000546.

[35] J. Venskus and P. Treigys. "Meteorological data influence on missing Vessel type detection using deep Multi-Stacked LSTM neural network". In: *Computer data analysis and modeling: stochastics and data science : proceedings of the XII international conference, Minsk, September 18-22, 2019*. Minsk: Minsk : Belarusian State University, 2019, p. 307–310. ISBN: 9789855668115.

[36] A.L. Ellefsen et al. "Online Fault Detection in Autonomous Ferries: Using Fault-type Independent Spectral Anomaly Detection". In: *IEEE Transactions on Instrumentation and Measurement* I.May (2020), pp. 1–1. ISSN: 0018-9456. DOI: 10.1109/tim.2020.2994012.

[37] H. Tang et al. "Detection of Abnormal Vessel Behaviour Based on Probabilistic Directed Graph Model". In: *Journal of Navigation* (2020), p. 1014. ISSN: 14697785. DOI: 10.1017/S0373463320000144.

[38] X. Shi and D.Y. Yeung. *Machine Learning for Spatiotemporal Sequence Forecasting: A Survey*. 2018. URL: http://arxiv.org/abs/1808.06865.

[39] M. Ranzato et al. *Video (language) modeling: a baseline for generative models of natural videos*. 2014. URL: http://arxiv.org/abs/1412.6604.

[40] A. Zonoozi et al. “Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns”. In: *IJCAI International Joint Conference on Artificial Intelligence* 2018-July (2018), pp. 3732–3738. ISSN: 10450823. DOI: 10.24963/ijcai.2018/519.

[41] X. Shi et al. “Convolutional LSTM network: A machine learning approach for precipitation nowcasting”. In: *Advances in Neural Information Processing Systems* 2015-Janua (2015), pp. 802–810. ISSN: 10495258.

[42] R. Yu et al. “Deep learning: A generic approach for extreme condition traffic forecasting”. In: *Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017* (2017), pp. 777–785. DOI: 10.1137/1.9781611974973.87.

[43] R. Senanayake, S. O’callaghan, and F. Ramos. “Predicting spatio-temporal propagation of seasonal influenza using variational Gaussian process regression”. In: *30th AAAI Conference on Artificial Intelligence, AAAI 2016* (2016), pp. 3901–3907.

[44] Y. Zheng et al. “Forecasting Fine-Grained Air Quality Based on Big Data”. In: *Proceedings of the 21th SIGKDD conference on Knowledge Discovery and Data Mining*. KDD 2015, Aug. 2015. URL: https://www.microsoft.com/en-us/research/publication/forecasting-fine-grained-air-quality-based-on-big-data/.

[45] A. Safikhani et al. “Spatio-temporal modeling of yellow taxi demands in New York City using generalized STAR models”. In: *International Journal of Forecasting* 36.3 (2020), pp. 1138–1148. ISSN: 01692070. DOI: 10.1016/j.ijforecast.2018.10.001.

[46] A. Alahi et al. “Social LSTM: Human Trajectory Prediction in Crowded Spaces”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 961–971.

[47] K. Yamaguchi et al. “Who are you with and where are you going?” In: *CVPR 2011*. 2011, pp. 1345–1352.

[48] A. Rudenko et al. “Human Motion Trajectory Prediction: A Survey”. In: *Journal of Vibration and Control* 1.May (May 2019). DOI: 10.1177/0278364920917446. URL: http://arxiv.org/abs/1905.06113%20http://dx.doi.org/10.1177/0278364920917446.

[49] K. Fragkiadaki et al. “Recurrent network models for human dynamics”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 4346–4354. ISSN: 15505499. DOI: 10.1109/ICCV.2015.494.

[50] J. Isensee, G. Datseris, and U. Parlitz. "Predicting Spatio-temporal Time Series Using Dimension Reduced Local States". In: *Journal of Nonlinear Science* 30.3 (2020), pp. 713–735. ISSN: 14321467. DOI: 10.1007/s00332-019-09588-7. URL: https://doi.org/10.1007/s00332-019-09588-7.

[51] R. Asadi and A.C. Regan. "A spatio-temporal decomposition based deep neural network for time series forecasting". In: *Applied Soft Computing Journal* 87 (2020). ISSN: 15684946. DOI: 10.1016/j.asoc.2019.105963.

[52] N. Cruz, L.G. Marin, and D. Saez. "Prediction Intervals With LSTM Networks Trained By Joint Supervision". In: *IJCNN. International Joint Conference on Neural Networks. Budapest, Hungary* 14-19 July 2019 (2019).

[53] Danish Maritime Authority. *Historical AIS data.* https://www.dma.dk/SikkerhedTilSoes/Sejladsinformation/AIS/Sider/default.aspx. 2020.

[54] World Weather Online. *World Weather Online meteorological data of Danish waters region.* https://www.worldweatheronline.com/. 2020.

[55] Gintautas Dzemyda, Olga Kurasova, and Julius Žilinskas. *Multidimensional Data Visualization: Methods and Applications.* Vol. 75. Springer Science & Business Media, 2012.

[56] Maria Riveiro et al. "Supporting maritime situation awareness using self organizing maps and gaussian mixture models". In: *Frontiers in Artificial Intelligence and Applications* 173 (2008), p. 84.

[57] J. Venskus et al. "Detecting Maritime traffic anomalies with long-short term memory recurrent neural network". In: *11th international workshop on data analysis methods for software systems (DAMSS 2019), Druskininkai, Lithuania, November 28-30, 2019.* Vilnius: Lithuanian Computer Society, Vilnius University Institute of Data Science and Digital Technologies, Lithuanian Academy of Sciences. Vilnius : Vilnius University Press, 2019, p. 89. ISBN: 9786090703243. DOI: 10.15388/Proceedings.2019.8..

[58] P.N.R. Chopde and M.K. Nichat. "Landmark Based Shortest Path Detection by Using A* and Haversine Formula". In: *Department of Computer Science and Engineering, G.H. Raisoni College of Engineering and Management* 1 (2013), p. 2.

[59] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: `http://jmlr.org/papers/v15/srivastava14a.html`.

[60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[61] N. V. Chawla et al. “SMOTE: Synthetic Minority Over-sampling Technique”. In: *Journal of Artificial Intelligence Research* 2009.Sept. 28 (June 2011), pp. 321–357. ISSN: 10769757. DOI: `10.1613/jair.953`. URL: `https://arxiv.org/pdf/1106.1813.pdf%20http://arxiv.org/abs/1106.1813%20http://dx.doi.org/10.1613/jair.953`.

[62] S. Alla and S. K. Adar. *Beginning Anomaly Detection Using Python-Based Deep Learning*. Vol. 75. Apress, Berkeley, CA, 2019. ISBN: 978-1-4842-5176-8.

[63] P.K. Diederik and B. Jimmy. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: `http://arxiv.org/abs/1412.6980`.

[64] Olga Kurasova and Alma Molytė. “Quality of quantization and visualization of vectors obtained by neural gas and self-organizing map”. In: *Informatica* 22.1 (2011), pp. 115–134.

[65] Teuvo Kohonen et al. “Engineering applications of the self-organizing map”. In: *Proceedings of the IEEE* 84.10 (1996), pp. 1358–1384.

[66] Juha Vesanto et al. “SOM toolbox for Matlab 5”. In: *Helsinki University of Technology, Finland* (2000).

[67] Yonggang Liu, Robert H Weisberg, and Christopher NK Mooers. “Performance evaluation of the self-organizing map for feature extraction”. In: *Journal of Geophysical Research: Oceans* 111.C5 (2006).

[68] Gintautas Dzemyda. “Visualization of a set of parameters characterized by their correlation matrix”. In: *Computational statistics & data analysis* 36.1 (2001), pp. 15–30.

[69] Teuvo Kohonen. “Self-organized formation of topologically correct feature maps”. In: *Biological cybernetics* 43.1 (1982), pp. 59–69.

[70] Pavel Stefanovic and Olga Kurasova. “Investigation on Learning Parameters of Self-Organizing Maps”. In: *Baltic Journal of Modern Computing* 2.2 (2014), p. 45.

[71] Pavel Stefanovič and Olga Kurasova. "Influence of Learning Rates and Neighboring Functions on Self-Organizing Maps". In: *International Workshop on Self-Organizing Maps*. Springer. 2011, pp. 141–150.

[72] Ding Yingying, He Yan, and Jiang Jingping. "Multi-robot cooperation method based on the ant algorithm". In: *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE. 2003, pp. 14–18.

[73] J. Venskus et al. "Self-learning adaptive algorithm for maritime traffic abnormal movement detection based on virtual pheromone method". In: *Proceedings of the 2015 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2015 - Part of SummerSim 2015 Multiconference*. Vol. 47. 2015. ISBN: 9781510810600. DOI: 10.1109/SPECTS.2015.7285281.

[74] M. Riveiro, G. Falkman, and T. Ziemke. "Improving maritime anomaly detection and situation awareness through interactive visualization". In: *Information Fusion, 2008 11th International Conference on*. IEEE. 2008, pp. 1–8.

[75] H. Gunes Kayacik, A. Nur Zincir-Heywood, and Malcolm I. Heywood. "A hierarchical SOM-based intrusion detection system". In: *Engineering Applications of Artificial Intelligence* 20.4 (2007), p. 439. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2006.09.005. URL: http://www.sciencedirect.com/science/article/pii/S0952197606001606.

[76] Z. Wang, J. Lin, and Z. Wang. "Accelerating Recurrent Neural Networks: A Memory-Efficient Approach". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.10 (2017), pp. 2763–2775. DOI: 10.1109/TVLSI.2017.2717950.

[77] Amir H. Jafari and Martin T. Hagan. "Application of new training methods for neural model reference control". In: *Engineering Applications of Artificial Intelligence* 74 (2018), pp. 312–321. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2018.07.005. URL: http://www.sciencedirect.com/science/article/pii/S0952197618301490.

[78] B. Cannas et al. "Disruption prediction with adaptive neural networks for ASDEX Upgrade". In: *Fusion Engineering and Design* 86.6 (2011). Proceedings of the 26th Symposium of Fusion Technology (SOFT-26), pp. 1039–1044. ISSN: 0920-3796. DOI: https://doi.org/10.1016/j.fusengdes.2011.01.069. URL: http://www.sciencedirect.com/science/article/pii/S0920379611000810.

[79] M. He and D. He. “Deep Learning Based Approach for Bearing Fault Diagnosis”. In: *IEEE Transactions on Industry Applications* 53.3 (2017), pp. 3057–3065. DOI: 10.1109/TIA.2017.2661250.

[80] Jolita Bernataviciene et al. “Method for Visual Detection of Similarities in Medical Streaming Data”. In: *International Journal of Computers, Communications & Control (IJCCC)* 10 (Feb. 2015), pp. 8–21. DOI: 10.15837/ijccc.2015.1.1310.

[81] J. Venskus and P. Treigys. “Preparation of training data by filling in missing vessel type data using deep multi-stacked LSTM neural network for abnormal marine transport evaluation”. In: *ITISE 2019:International Conference on Time Series and Forecasting: proceedings of abstracts. Granada, Spain, September, 25-27, 2019.* Granada : Universidad de Granada, 2019, p. 38. ISBN: 978841-7970796.

[82] J. Venskus, P. Treigys, and J Markevičiūtė. “Unsupervised Marine Vessel Trajectory Prediction using LSTM Network and Wild Bootstrapping Techniques”. In: *Nonlinear Analysis: Modelling and Control* ??? (2020).

[83] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[84] C. Chiu et al. “State-of-the-art speech recognition with sequence-to-sequence models”. In: *CoRR* abs/1712.01769.8 (2017). URL: http://arxiv.org/abs/1712.01769.

[85] A. Graves and J. Schmidhuber. “Offline handwriting recognition with multidimensional recurrent neural networks”. In: *Advances in Neural Information Processing Systems* 21 (2009). URL: http://papers.nips.cc/paper/3449-offline-handwriting-recognition-%20with-multidimensional-recurrent-neural-networks.pdf.

[86] M.J. Hausknecht and P. Stone. “Deep recurrent q-learning for partially observable mdps”. In: *CoRR* abs/1507.06527 (2015). URL: http://arxiv.org/abs/1507.06527.

[87] M.A. Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE Journal* 37.2 (1991), pp. 233–243.

[88] Robertas Jurkus. “LSTM giliųjų neuroninių tinklų tyrimas laivo eigos prognozavimui naudojant didžiuosius eismo duomenis”. MA thesis. Klaipeda, Lithuania: Klaipėdos universitetas, 2020.

[89] L.G. Marin et al. “Prediction interval methodology based on fuzzy numbers and its extension to fuzzy systems and neural networks”. In: *Expert Systems with Applications* 119 (2019), pp. 128–141.

[90] N. Cruz, L.G. Marın, and D. Saez. “Neural network prediction interval based on joint supervision”. In: *2018 International Joint Conference on Neural Networks (IJCNN)* July 2018 (2018), pp. 1–8.

[91] V. Chew. “Confidence, Prediction, and Tolerance Regions for the Multivariate Normal Distribution”. In: *Journal of the American Statistical Association* 61.315 (1966), pp. 605–617.

[92] Richard Socher et al. “Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection”. In: *Advances in Neural Information Processing Systems* 24 (Jan. 2011).

[93] M. Cuturi and Blondel M. “Soft-DTW: a Differentiable Loss Function for Time-Series”. In: *Thirty-eighth International Conference on Machine Learning, ICML*. 2017.

[94] Cyril Goutte et al. “On Clustering fMRI Time Series”. In: *NeuroImage* 9.3 (1999), pp. 298–310. ISSN: 1053-8119. DOI: `https://doi.org/10.1006/nimg.1998.0391`. URL: `http://www.sciencedirect.com/science/article/pii/S1053811998903913`.

[95] F. Petitjean, A. Ketterlin, and P. Gancarski. “A global averaging method for dynamic time warping, with applications to clustering”. In: *Pattern Recognition, Elsevier* 44 (Mar. 2011), pp. 678–693.

A APPENDIX - Pair plot of numerical AIS features

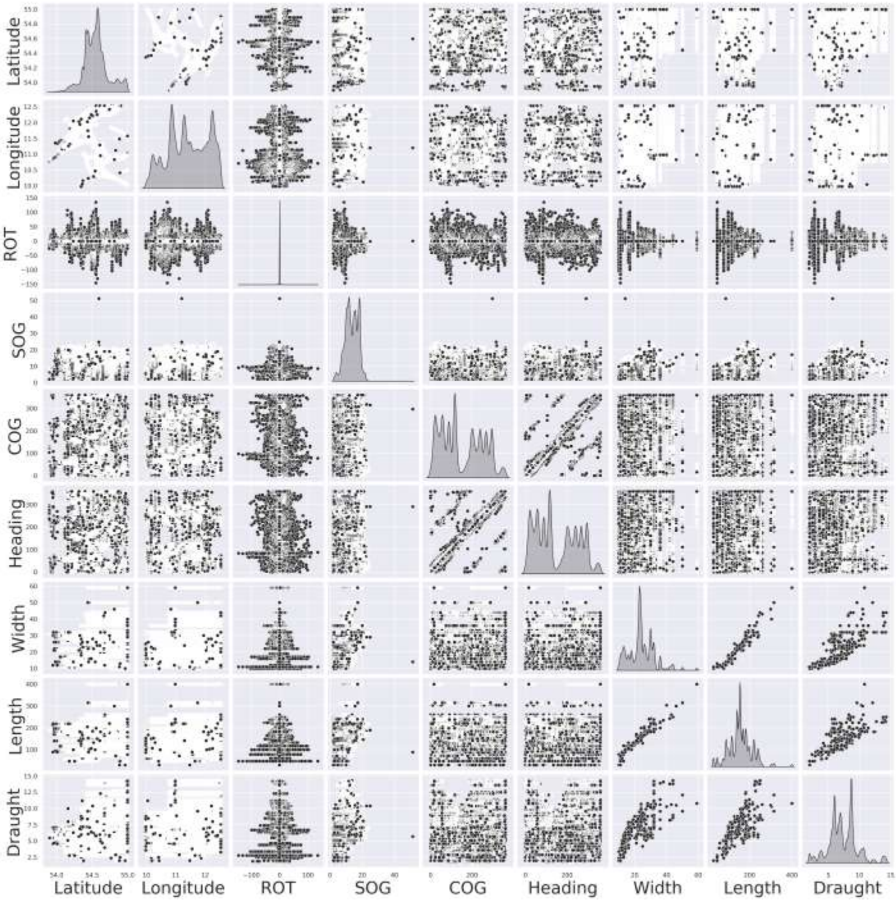
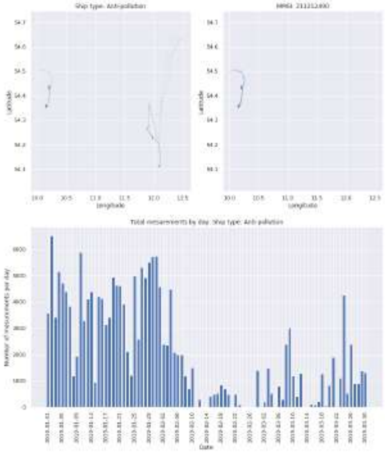
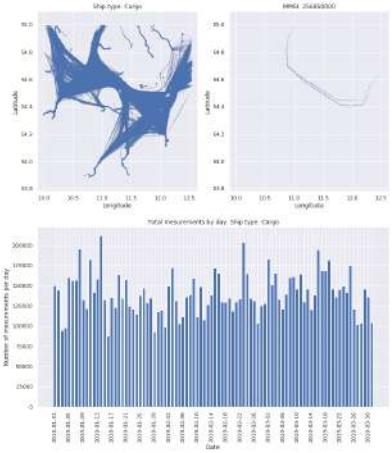
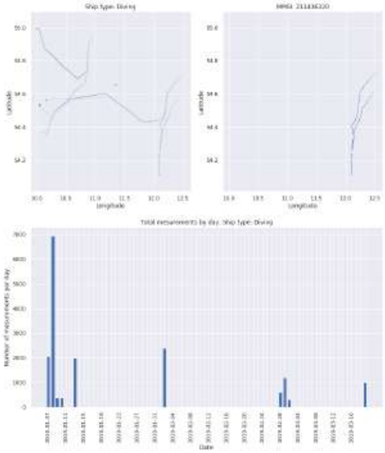


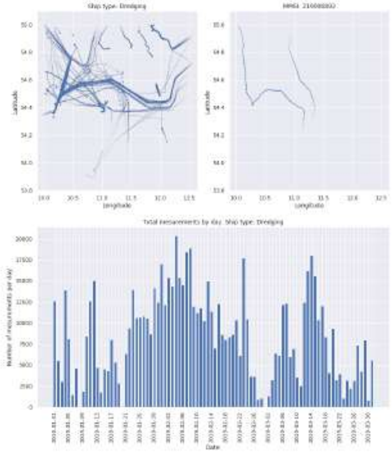
Figure 29: Pair plot of numerical features.

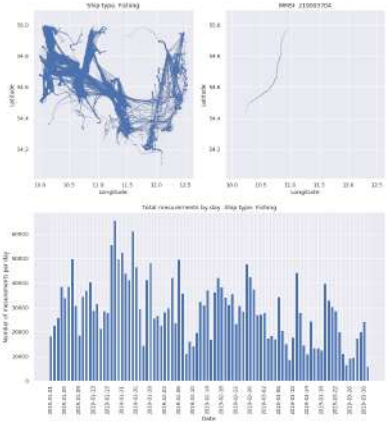
B APPENDIX - Vessel type visualizations

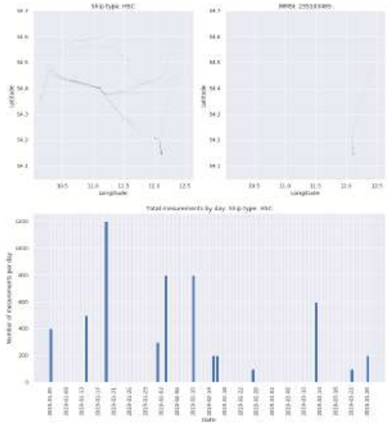


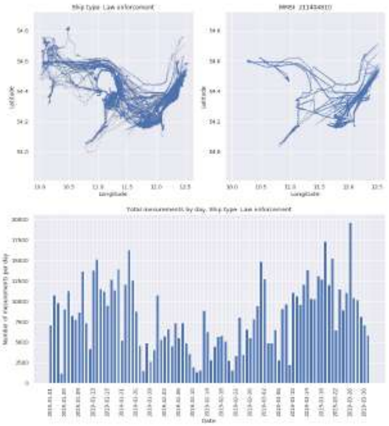


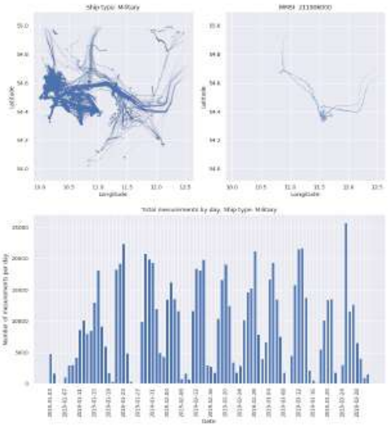


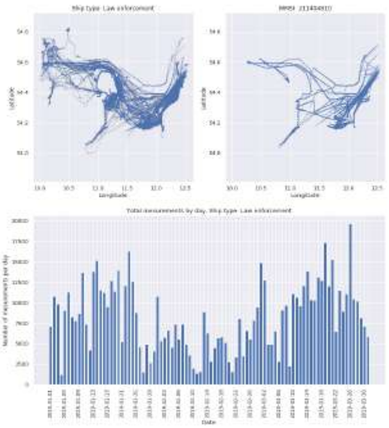


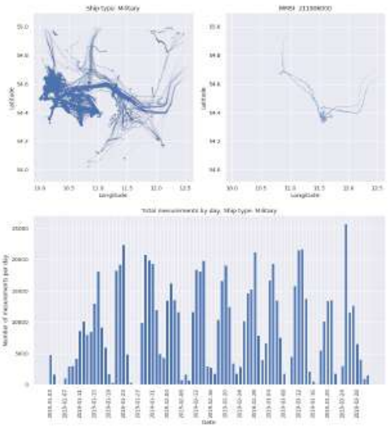

Ship type: Fishing
MMSI: 219003704
Latitude
Longitude
Total measurements by day, Ship type: Fishing
Number of measurements per day
Date

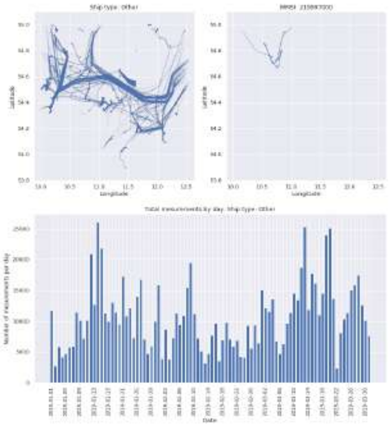

Ship type: HSC
MMSI: 255103485
Latitude
Longitude
Total measurements by day, Ship type: HSC
Number of measurements per day
Date

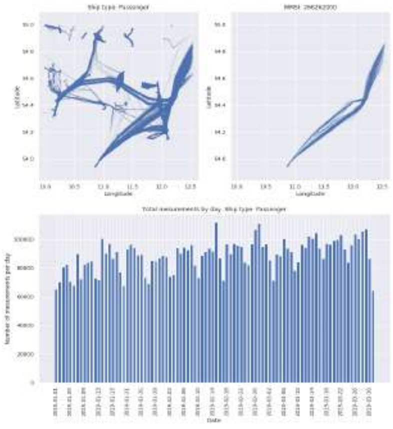

Ship type: Law enforcement
MMSI: 211404810
Latitude
Longitude
Total measurements by day, Ship type: Law enforcement
Number of measurements per day
Date

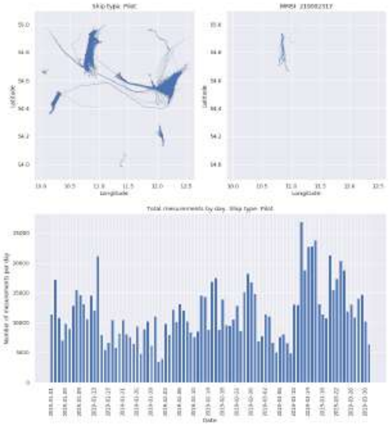

Ship type: Military
MMSI: 211906000
Latitude
Longitude
Total measurements by day, Ship type: Military
Number of measurements per day
Date

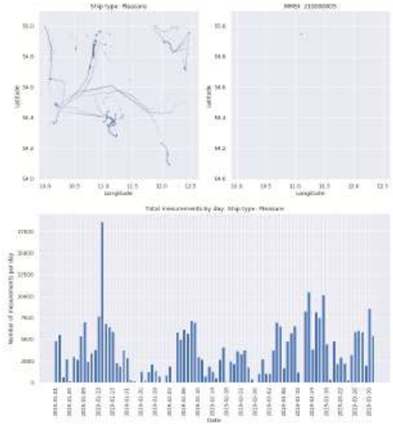

Ship type: Law enforcement
MMSI: 211404810
Latitude
Longitude
Total measurements by day, Ship type: Law enforcement
Number of measurements per day
Date

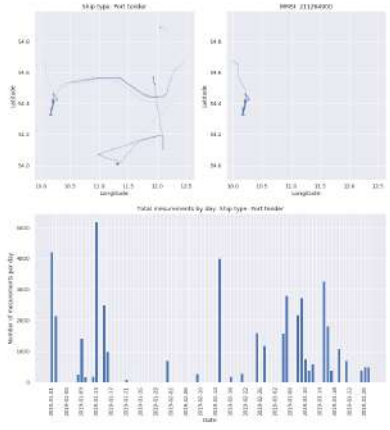

Ship type: Military
MMSI: 211906000
Latitude
Longitude
Total measurements by day, Ship type: Military
Number of measurements per day
Date

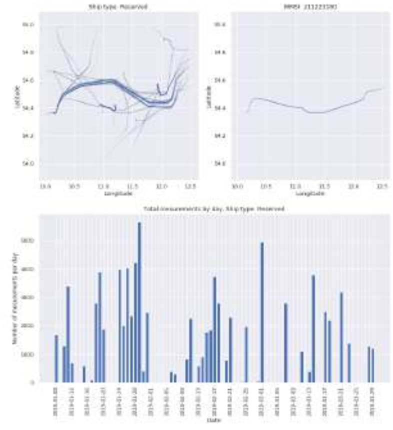

Ship type: Other
MMSI: 215887000
Latitude
Longitude
Total measurements by day. Ship type: Other
Number of measurements per day
Date

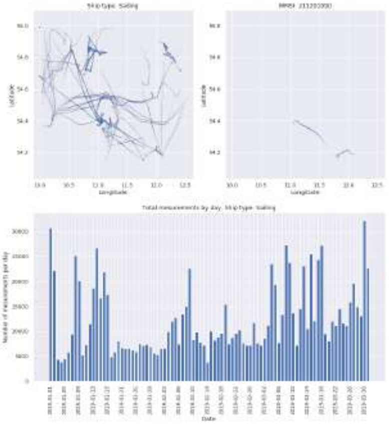

Ship type: Passenger
MMSI: 266262000
Latitude
Longitude
Total measurements by day. Ship type: Passenger
Number of measurements per day
Date

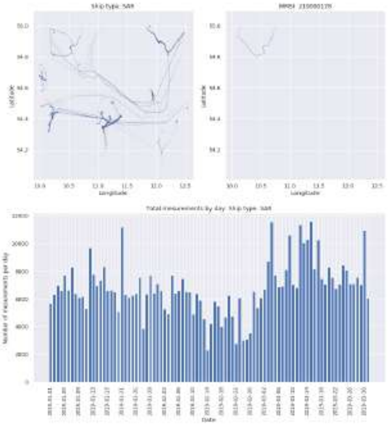

Ship type: Pilot
MMSI: 219002317
Latitude
Longitude
Total measurements by day. Ship type: Pilot
Number of measurements per day
Date

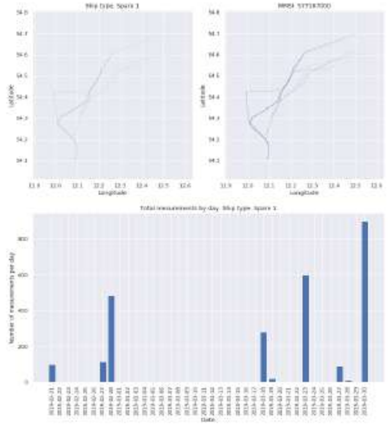

Ship type: Pleasure
MMSI: 219000805
Latitude
Longitude
Total measurements by day. Ship type: Pleasure
Number of measurements per day
Date

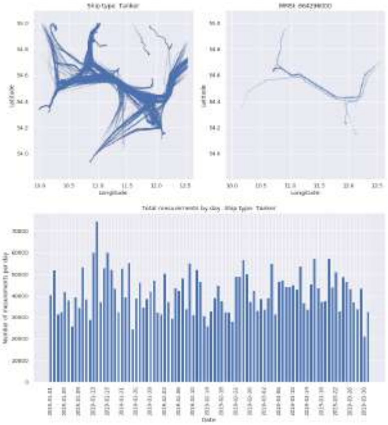

Ship type: Port tender
MMSI: 211264000
Latitude
Longitude
Total measurements by day. Ship type: Port tender
Number of measurements per day
Date

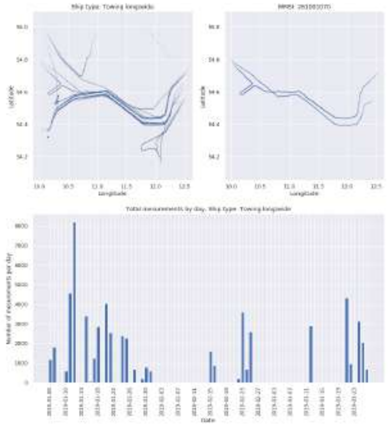

Ship type: Reserved
MMSI: 211223180
Latitude
Longitude
Total measurements by day. Ship type: Reserved
Number of measurements per day
Date

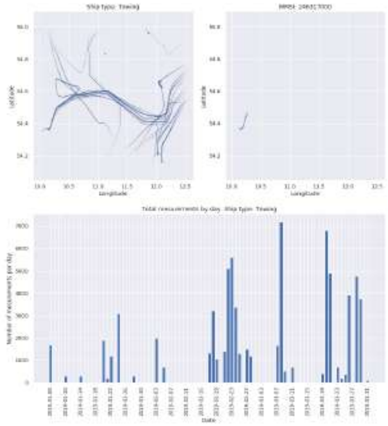

Ship type: Sailing
MMSI: 211201000
Latitude
Longitude
Total measurements by day. Ship type: Sailing
Number of measurements per day
Date

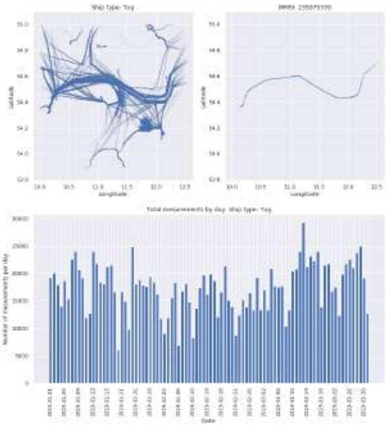

Ship type: SAR
MMSI: 219000179
Latitude
Longitude
Total measurements by day. Ship type: SAR
Number of measurements per day
Date

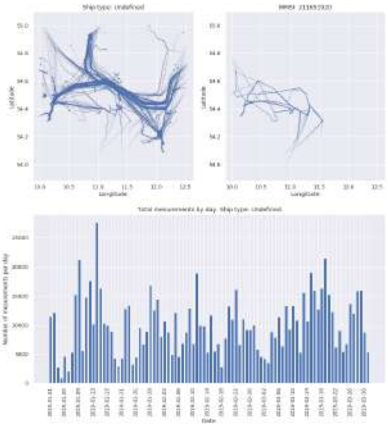

Ship type: Spare 1
MMSI: 577187000
Latitude
Longitude
Total measurements by day. Ship type: Spare 1
Number of measurements per day
Date


Ship type: Tanker
MMSI: 664298000
Latitude
Longitude
Total measurements by day. Ship type: Tanker
Number of measurements per day
Date


Ship type: Towing long/wide
MMSI: 261001070
Latitude
Longitude
Total measurements by day. Ship type: Towing long/wide
Number of measurements per day
Date


Ship type: Towing
MMSI: 246317000
Latitude
Longitude
Total measurements by day. Ship type: Towing
Number of measurements per day
Date


Ship type: Tug
MMSI: 255875339
Latitude
Longitude
55.0
54.8
54.6
54.4
54.2
54.0
53.8
10.0
10.5
11.0
11.5
12.0
12.5
Total measurements by day. Ship type: Tug
Number of measurements per day
30000
25000
20000
15000
10000
5000
0
Date


Ship type: Undefined
MMSI: 211653920
Latitude
Longitude
55.0
54.8
54.6
54.4
54.2
54.0
10.0
10.5
11.0
11.5
12.0
12.5
Total measurements by day. Ship type: Undefined
Number of measurements per day
25000
20000
15000
10000
5000
0
Date

C APPENDIX - Data tables of missing vessel type recognition model

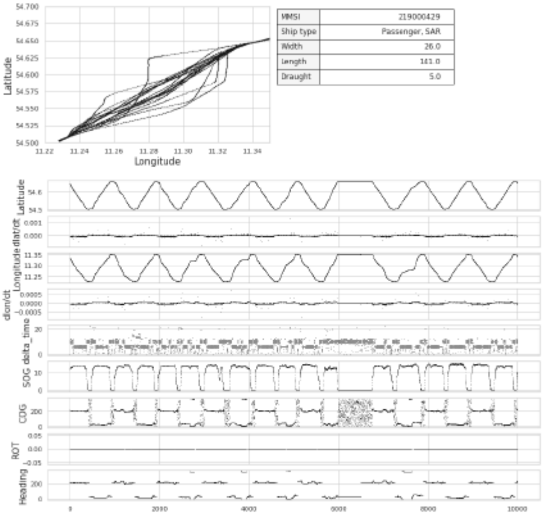
Table 35: Confusion matrix of vessel type prediction model

		True Class (FN)											
		Cargo	Tanker	Fishing	Passenger	Tug	Military	Sailing	Dredging	Pleasure	SAR	Pilot	Towing
Predicted class (FP)	Cargo	21488	201	5	237	187	0	0	0	3	0	40	33
	Tanker	217	21728	2	197	201	0	0	0	2	0	27	1
	Fishing	3	0	21390	11	108	2	84	22	33	0	155	341
	Passenger	180	144	4	21234	15	0	57	0	198	0	5	5
	Tug	104	1	171	10	19345	1	447	201	34	0	394	894
	Military	0	0	1	0	1	21633	2	0	0	107	0	0
	Sailing	0	0	74	29	398	1	20874	0	549	0	9	10
	Dredging	0	0	12	0	207	0	0	21636	0	0	66	207
	Pleasure	4	0	38	108	29	0	608	1	21271	0	2	1
	SAR	0	0	0	0	0	57	0	0	0	21956	0	0
	Pilot	75	25	178	13	401	1	18	48	14	0	20096	299
	Towing	25	4	117	3	1089	2	17	125	5	0	347	19616
	Reserved	0	0	1	2	4	2	3	4	1	0	1	3
	Law enfor.	0	0	0	1	0	409	0	1	0	48	0	1
	Towing long.	0	1	47	3	48	3	1	61	0	0	87	579
	HSC	0	0	0	259	0	0	0	0	0	0	0	0
	Port_ten.	15	7	71	4	78	0	0	12	1	0	877	101
	Diving	0	0	0	0	0	0	0	0	0	0	0	1
	Anti-pol.	0	0	0	0	0	0	0	0	0	0	5	17
	Spare_1	0	0	0	0	0	0	0	0	0	0	0	1
	WIG	0	0	0	0	0	0	0	0	0	0	0	1

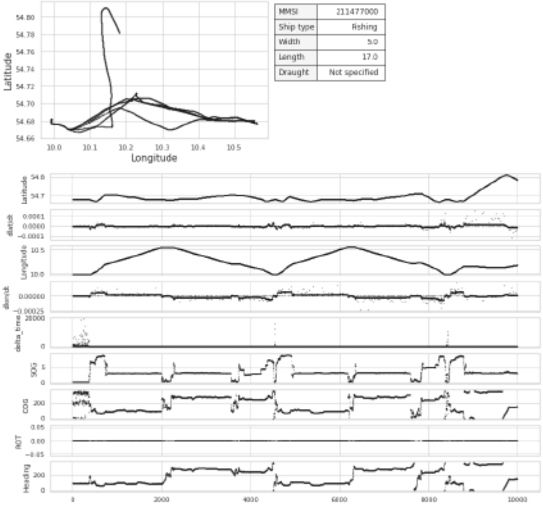
Table 36: Confusion matrix of vessel type prediction model (continued)

		True Class (FN)								
		Reserved	Law enfor.	Towing long.	HSC	Port tender	Diving	Anti-pol.	Spare 1	WIG
Predicted class (FP)	Tanker	0	0	3	0	21	0	0	0	0
	Cargo	0	0	4	0	17	0	0	0	0
	Fishing	0	0	143	0	51	0	0	0	0
	Passenger	1	0	1	155	1	0	0	0	0
	Tug	2	0	874	0	88	0	0	0	1
	Military	1	698	1	0	0	0	0	0	0
	Sailing	3	0	9	0	0	0	0	0	0
	Dredging	4	0	241	0	31	0	0	19	1
	Pleasure	2	0	4	0	0	0	0	0	0
	SAR	0	105	0	0	0	0	0	0	0
	Pilot	1	0	241	0	545	0	9	0	1
	Towing	1	0	1199	0	113	0	22	0	0
	Reserved	22095	0	0	0	0	0	0	0	2
	Law enforc.	1	20797	0	0	0	5	0	0	0
	Towing long.	0	0	19357	0	62	0	0	1	0
	HSC	0	0	0	21956	0	0	0	0	0
	Port tender	0	0	34	0	21182	0	0	0	0
	Diving	0	511	0	0	0	22106	0	0	0
	Anti-pol.	0	0	0	0	0	0	22080	0	0
	Spare 1	0	0	0	0	0	0	0	22091	1
	WIG	0	0	0	0	0	0	0	0	22105

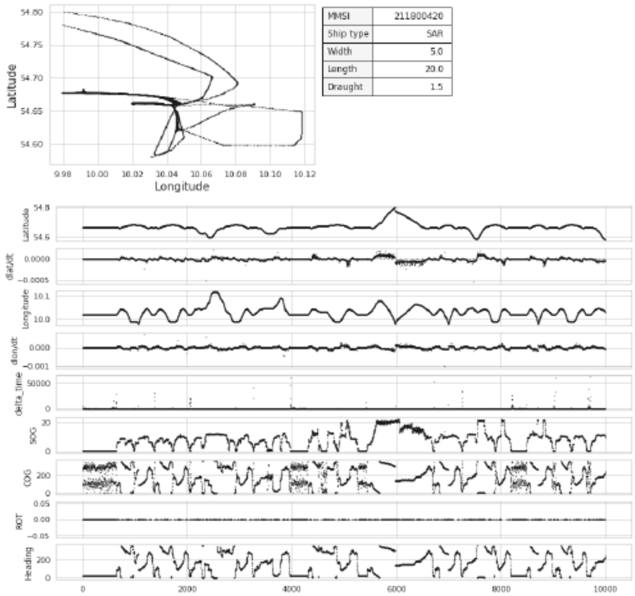
D APPENDIX - Random vessel track spatio temporal visualisation



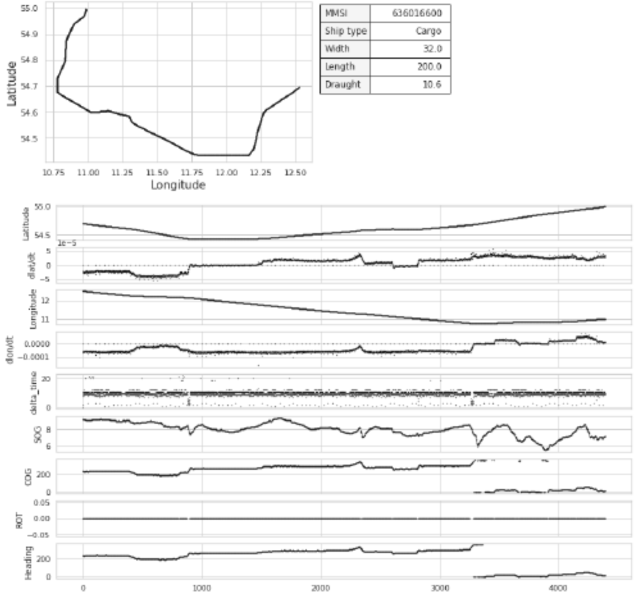
MMSI	219000429
Ship type	Passenger, SAR
Width	26.0
Length	141.0
Draught	5.0



MMSI	211477000
Ship type	Fishing
Width	5.0
Length	17.0
Draught	Not specified



MMSI	211800420
Ship type	SAR
Width	5.0
Length	20.0
Draught	1.5



MMSI	636016600
Ship type	Cargo
Width	32.0
Length	200.0
Draught	10.6

E APPENDIX - LSTM crisp model errors

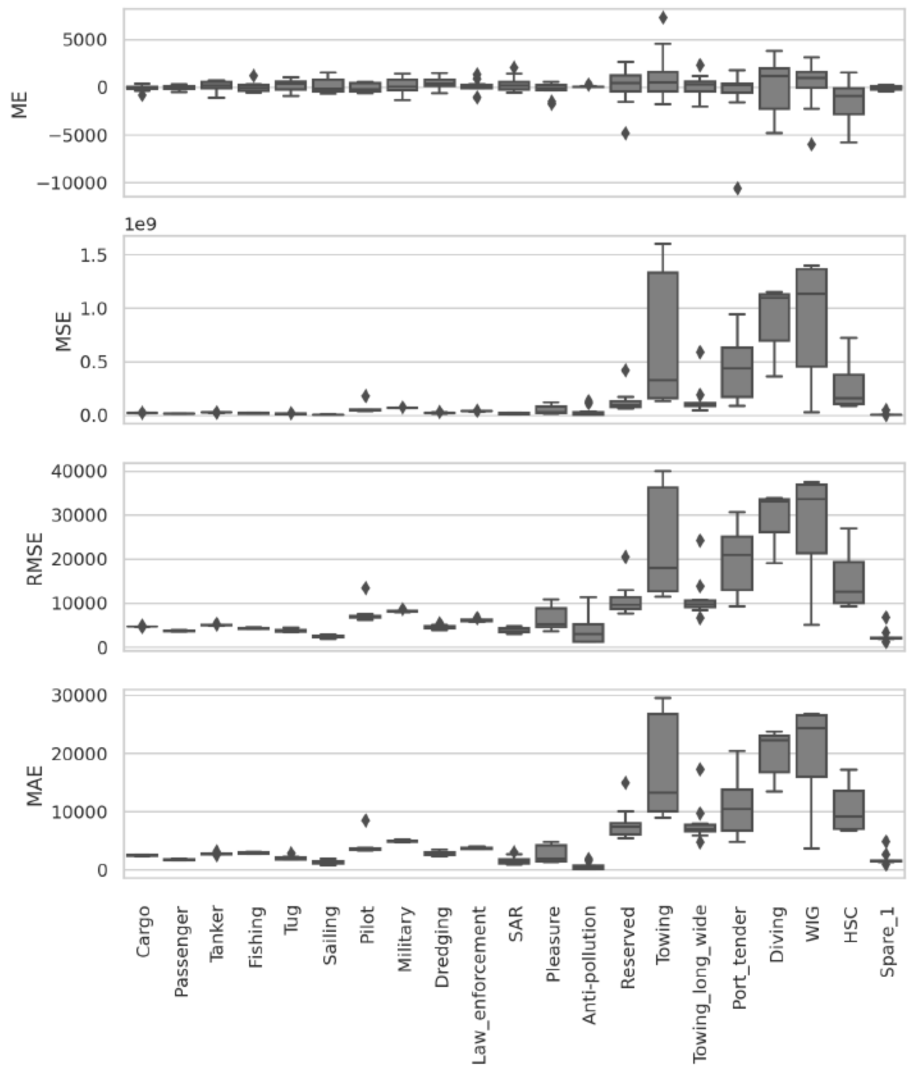


Figure 37: LSTM crisp model prediction errors

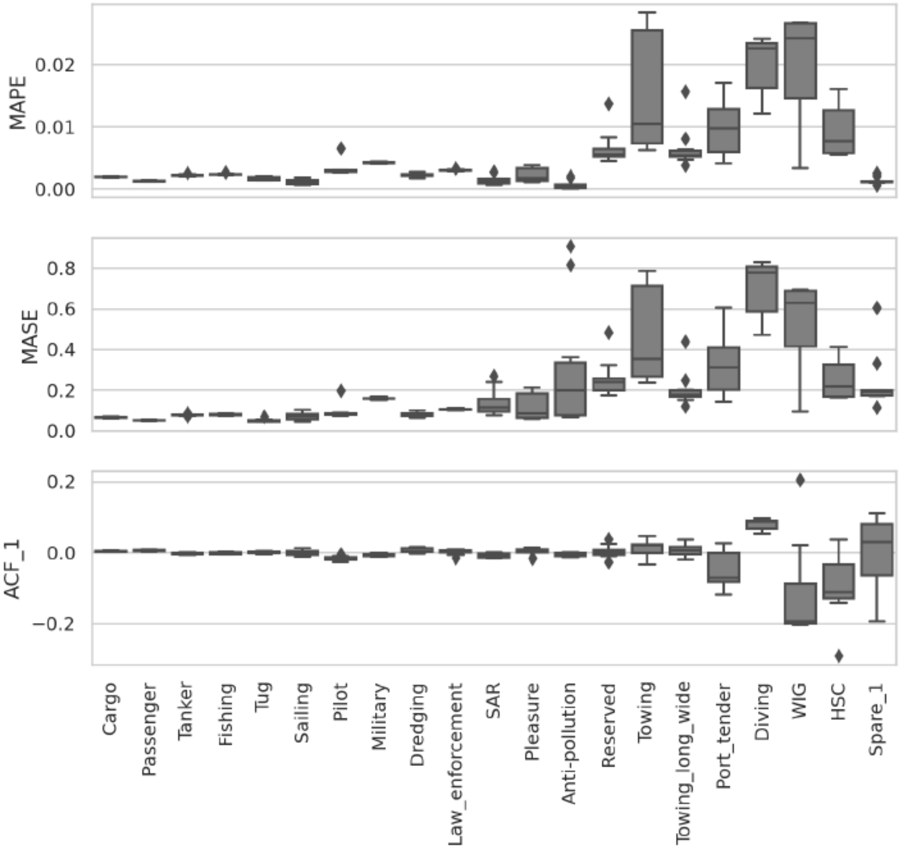


Figure 38: LSTM crisp model prediction errors (continued)

F APPENDIX - LSTM PICP and PINAW relation to lambda parameter

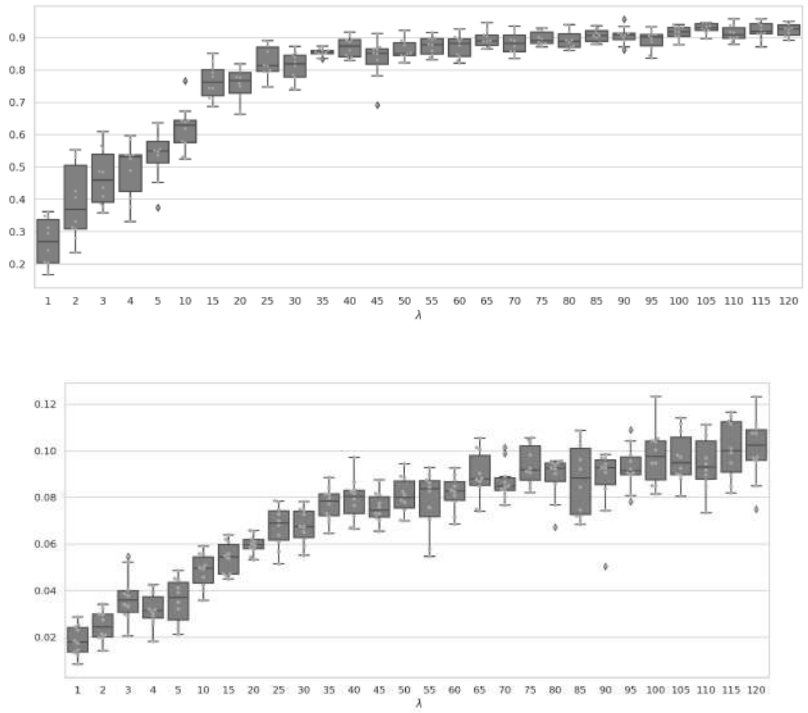


Figure 39: PICP and PINAW rate of "Cargo" ship type

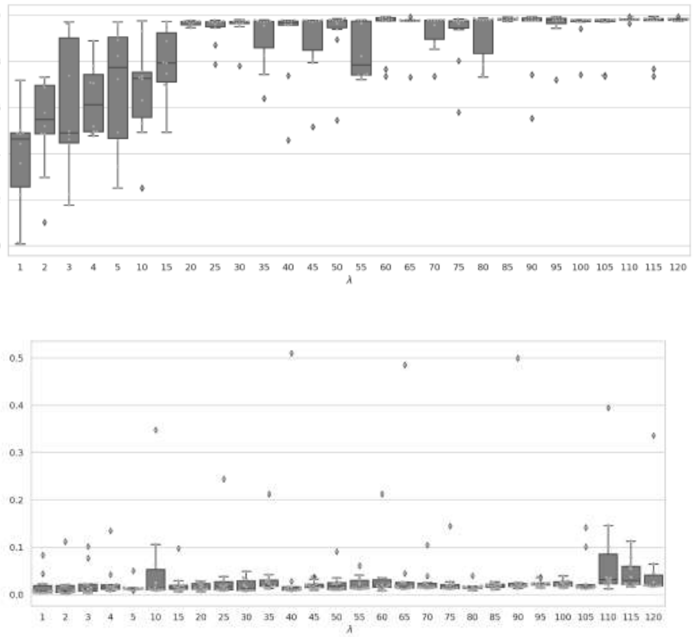


Figure 40: PICP and PINAW rate of "Anti-pollution" ship type

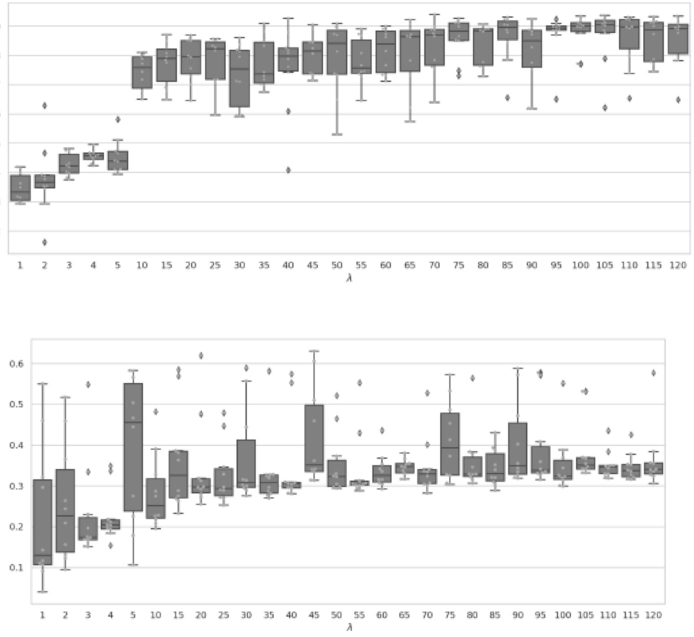


Figure 41: PICP and PINAW rate of "Diving" ship type

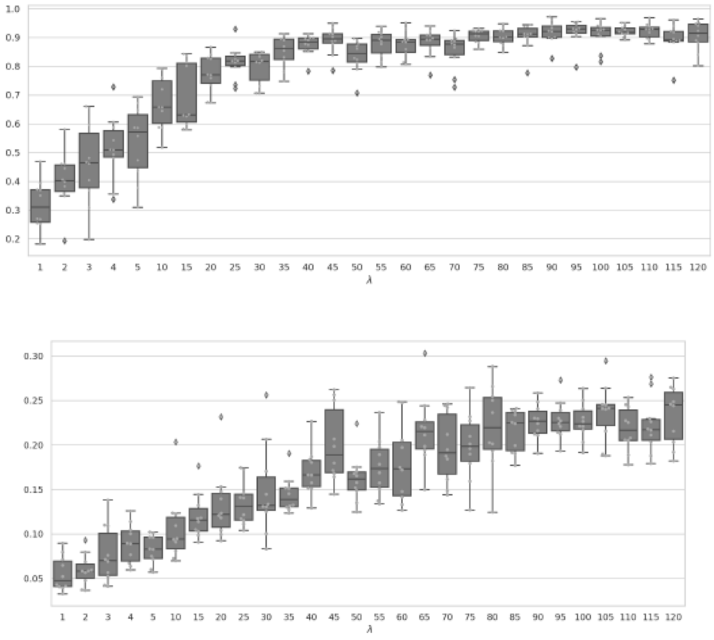


Figure 42: PICP and PINAW rate of "Dredging" ship type

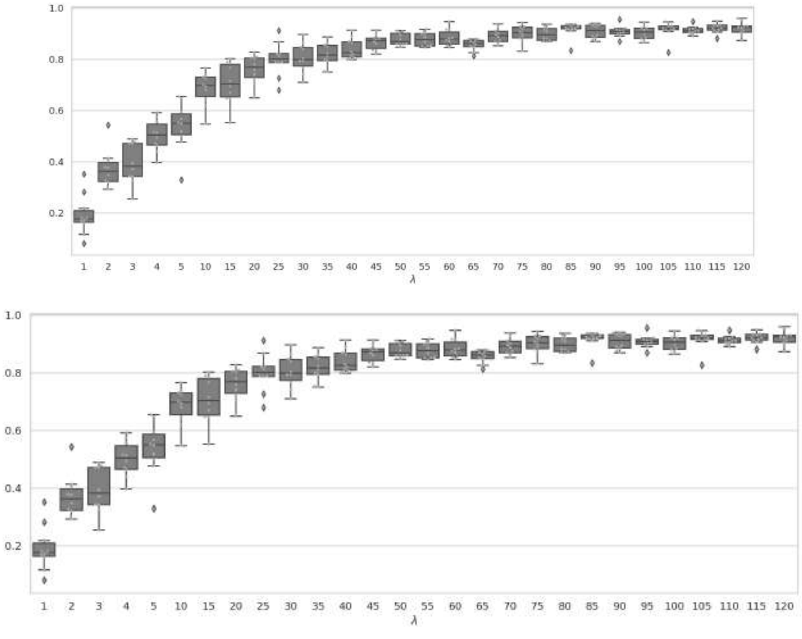


Figure 43: PICP and PINAW rate of "Fishing" ship type

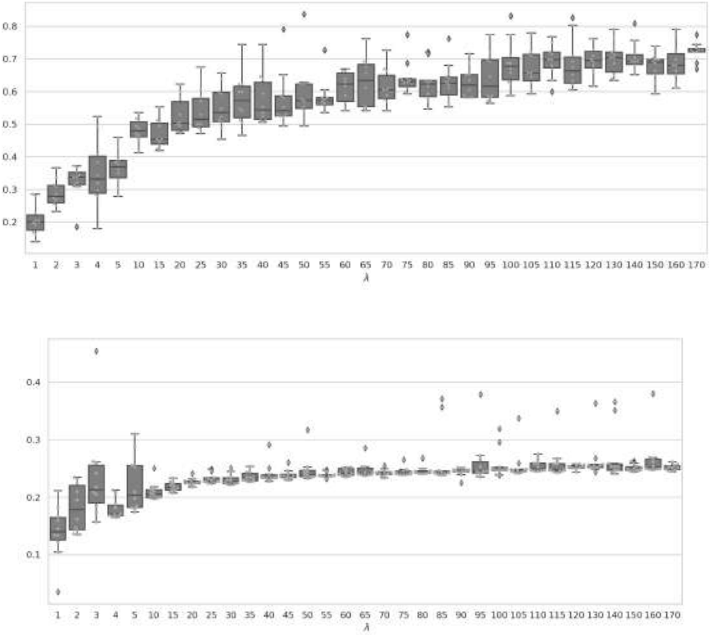


Figure 44: PICP and PINAW rate of "HSC" ship type

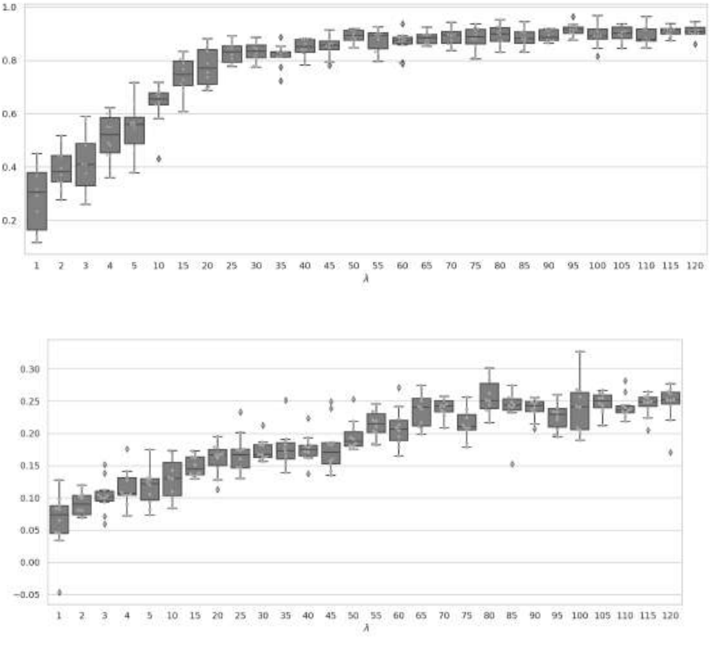


Figure 45: PICP and PINAW rate of "Law_enforcement" ship type

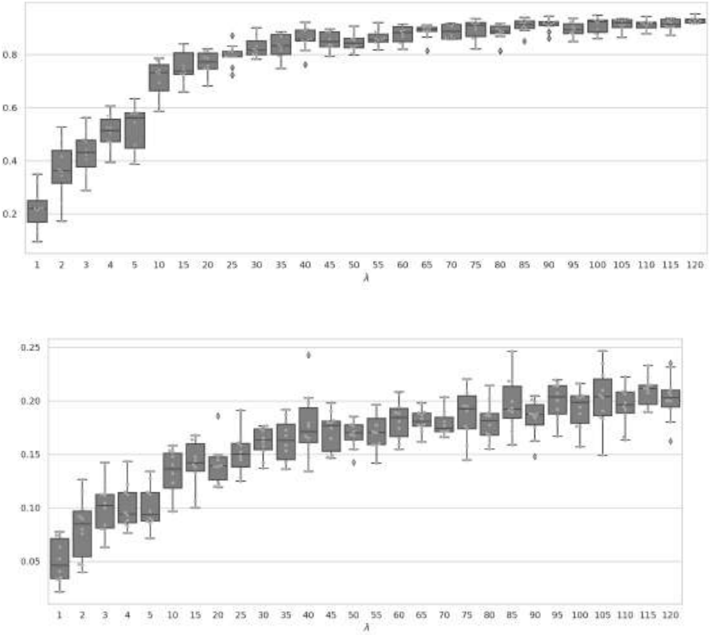


Figure 46: PICP and PINAW rate of "Military" ship type

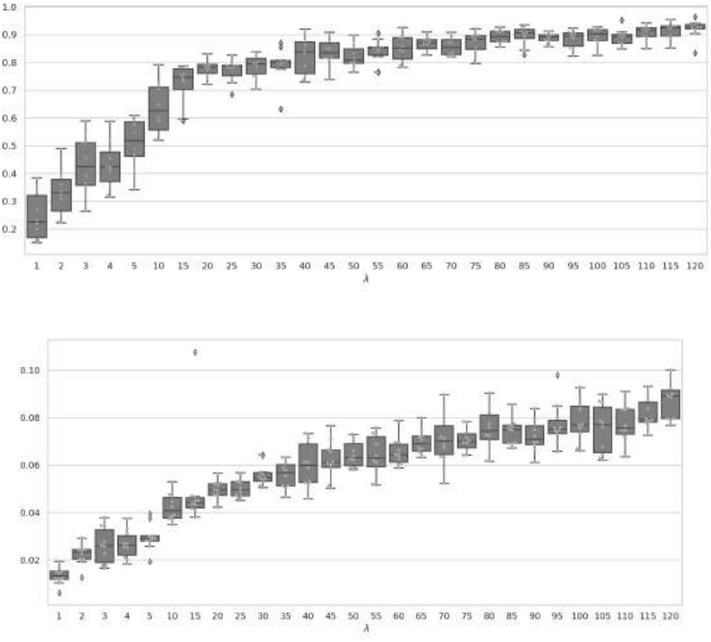


Figure 47: PICP and PINAW rate of "Passenger" ship type

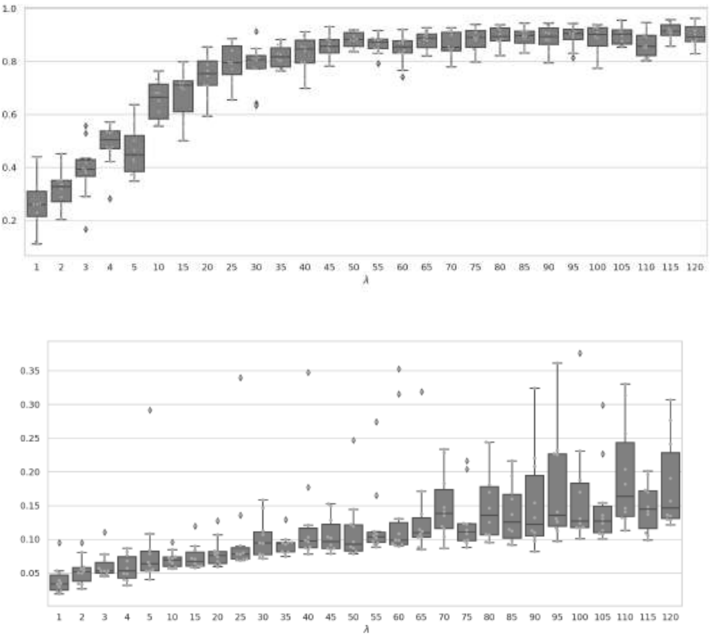


Figure 48: PICP and PINAW rate of "Pilot" ship type

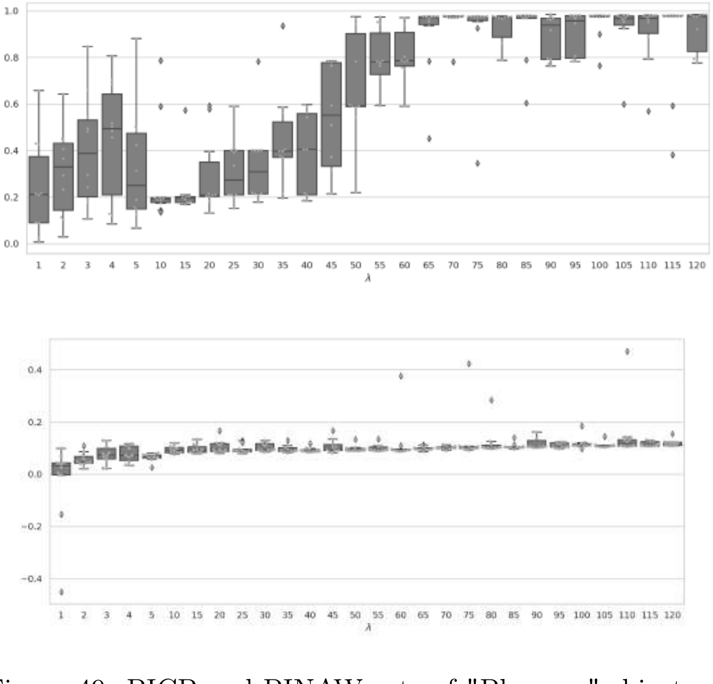


Figure 49: PICP and PINAW rate of "Pleasure" ship type

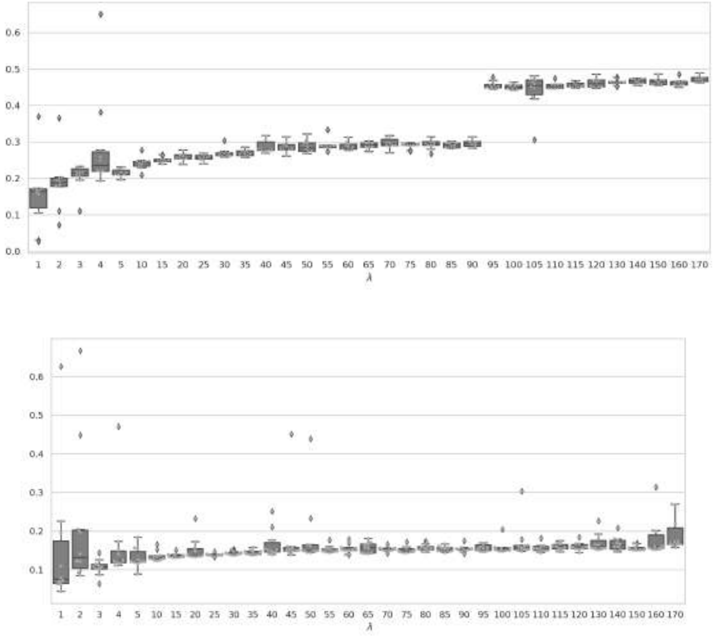


Figure 50: PICP and PINAW rate of "Port_tender" ship type

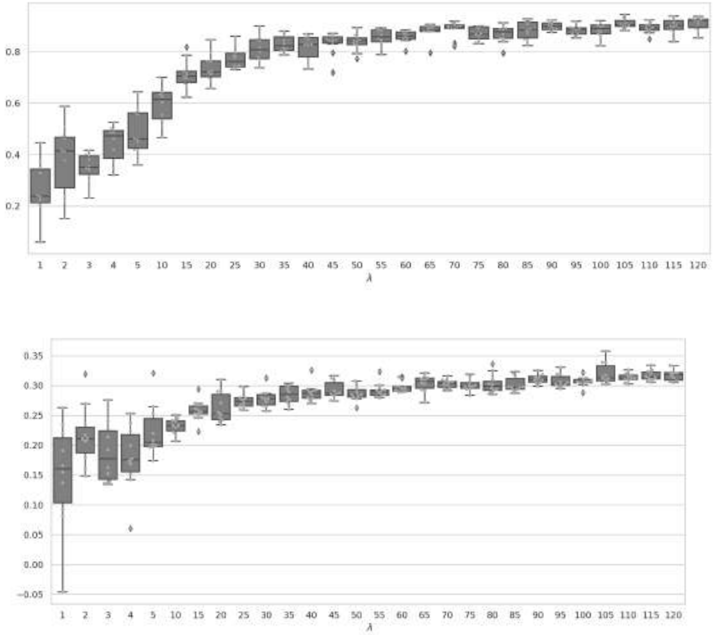


Figure 51: PICP and PINAW rate of "Reserved" ship type

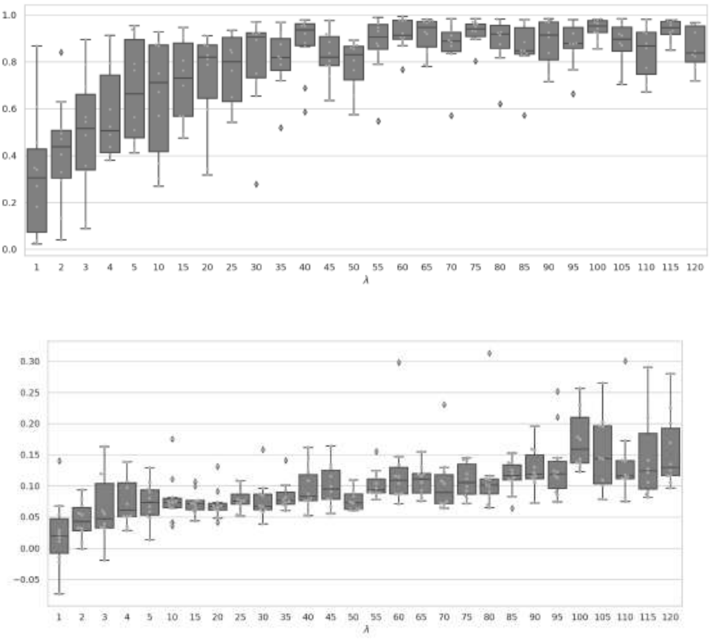


Figure 52: PICP and PINAW rate of "Sailing" ship type

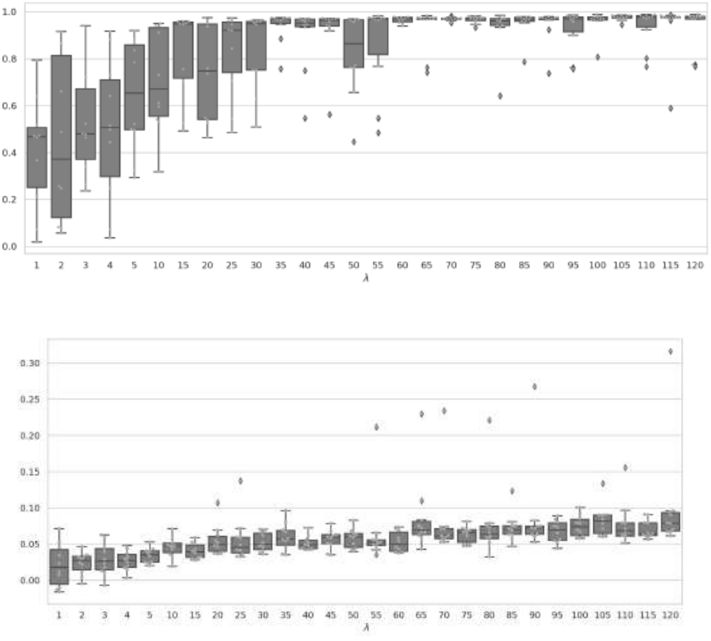


Figure 53: PICP and PINAW rate of "SAR" ship type

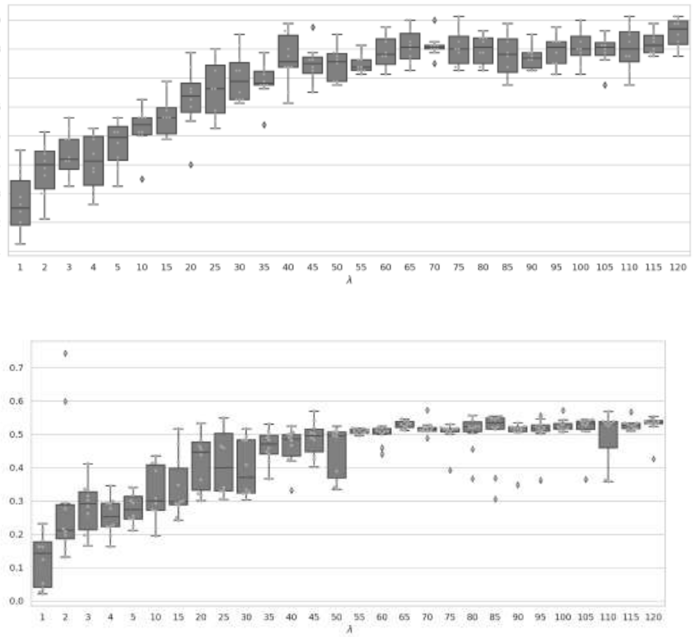


Figure 54: PICP and PINAW rate of "Spare_1" ship type

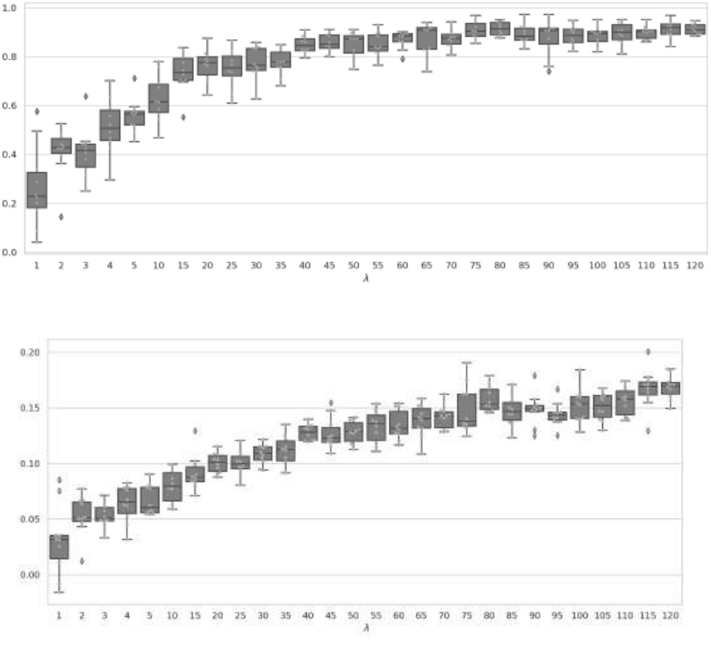


Figure 55: PICP and PINAW rate of "Tanker" ship type

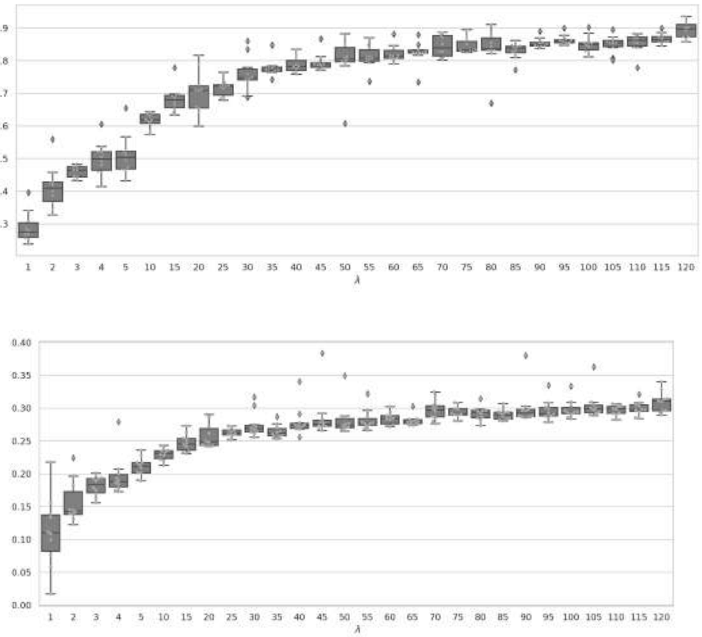


Figure 56: PICP and PINAW rate of "Towing_long_wide" ship type

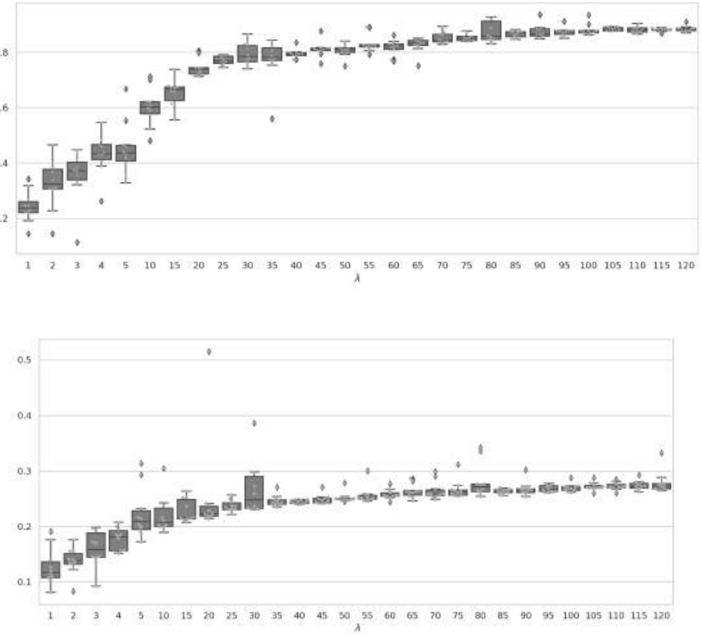


Figure 57: PICP and PINAW rate of "Towing" ship type

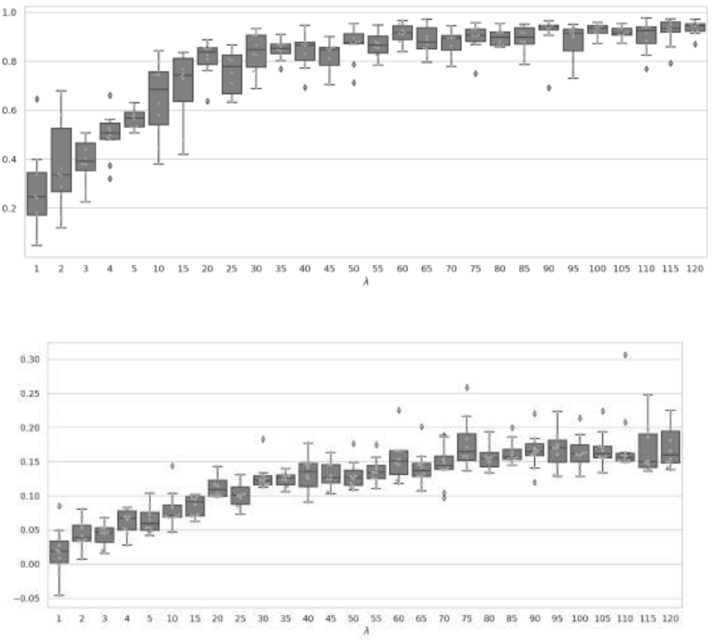


Figure 58: PICP and PINAW rate of "Tug" ship type

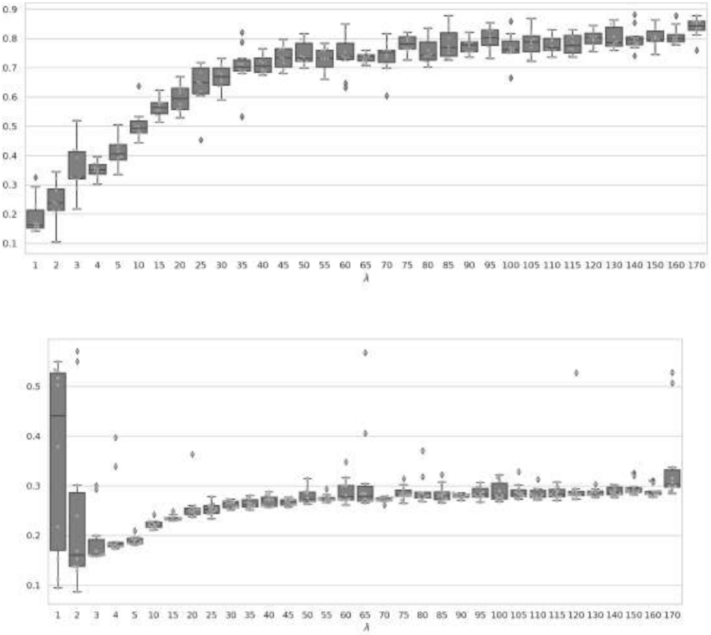
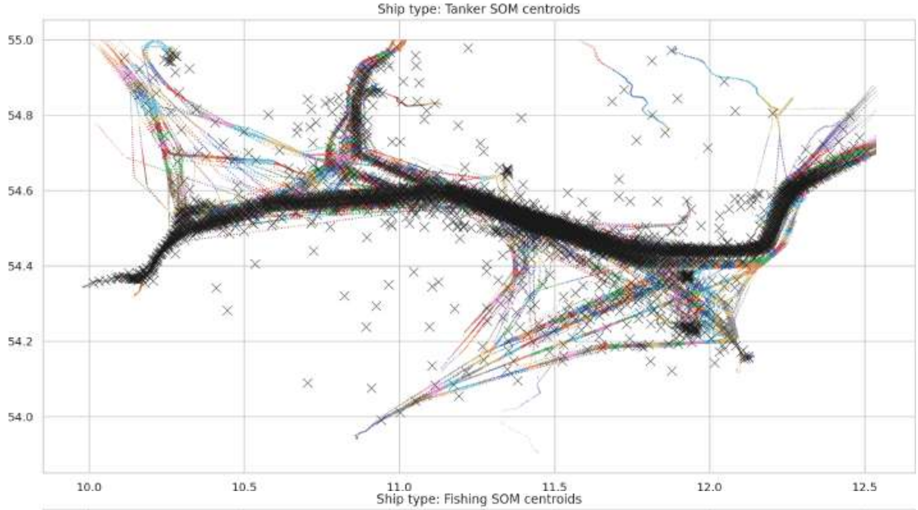
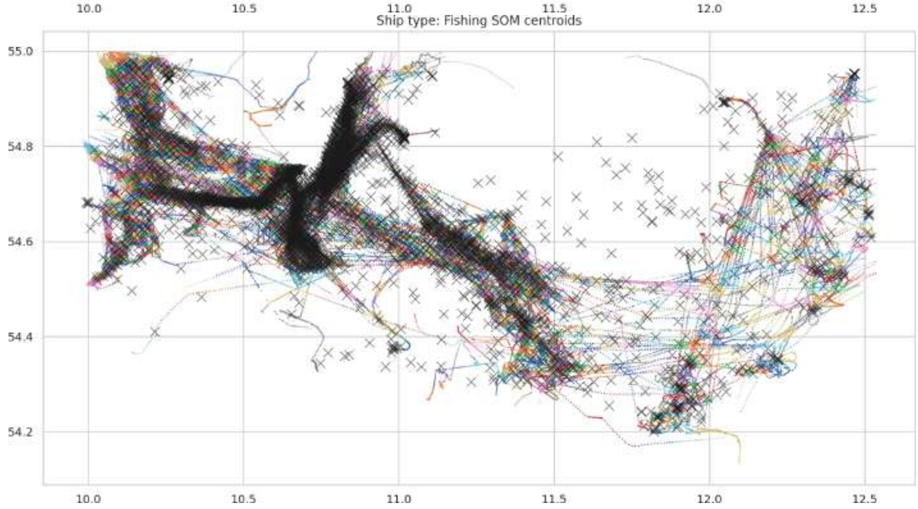


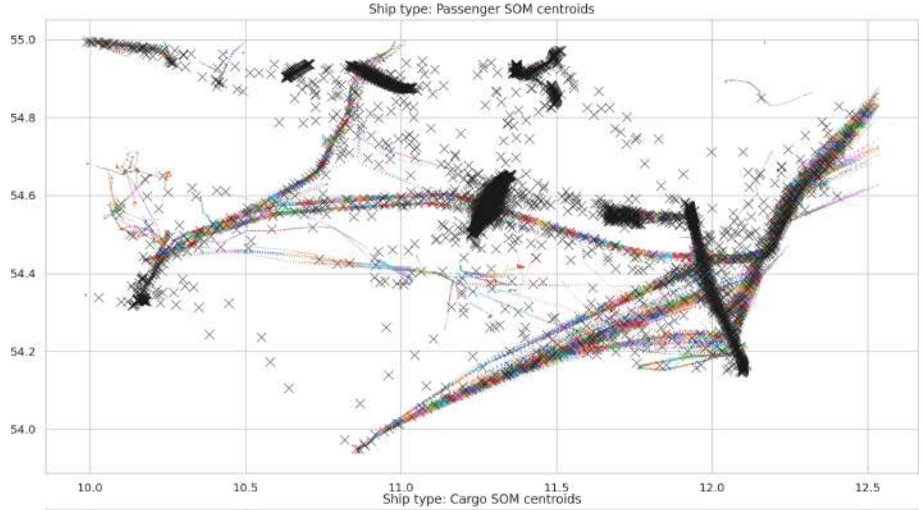
Figure 59: PICP and PINAW rate of "Wig" ship type

G APPENDIX - SOM and virtual pheromone figures

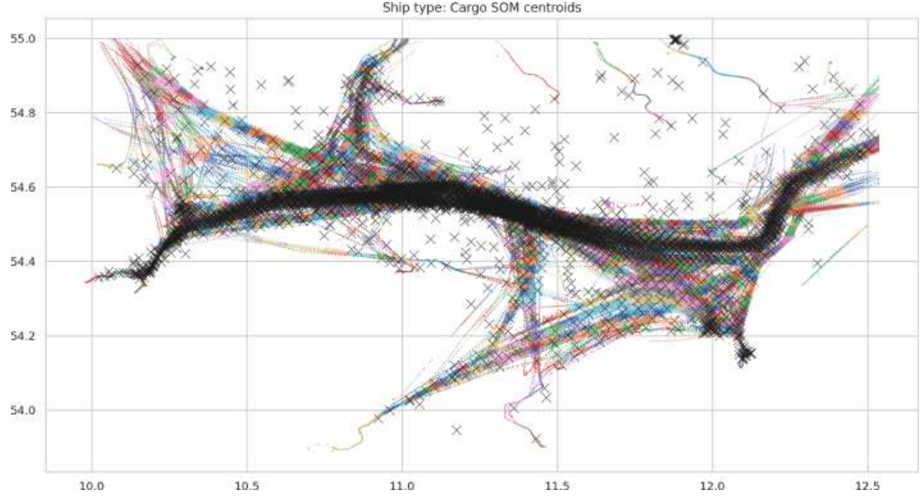


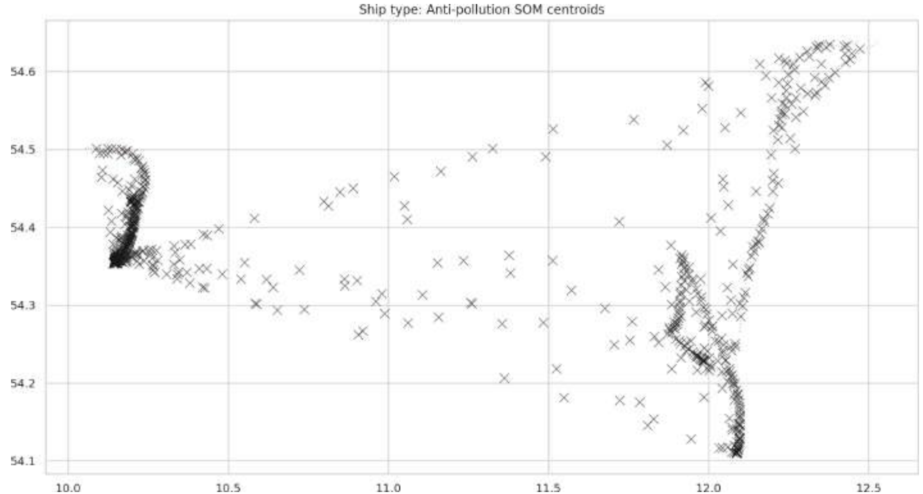


Ship type: Passenger SOM centroids

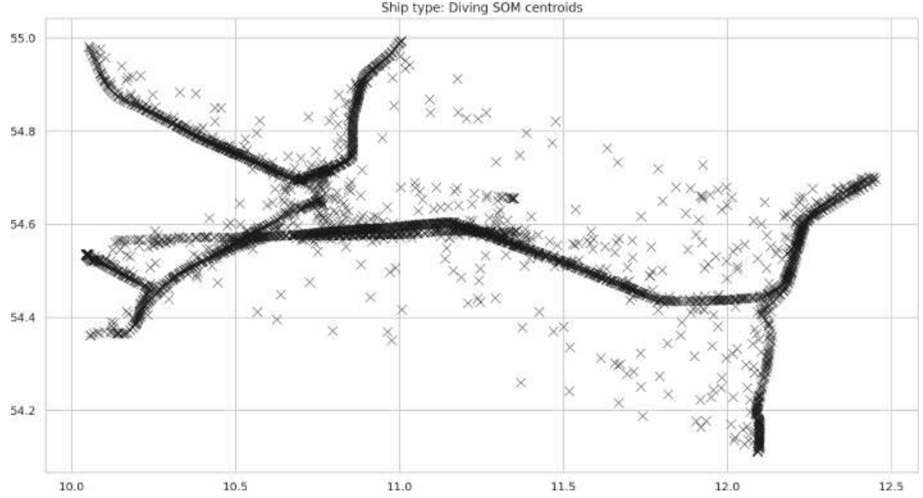


Ship type: Cargo SOM centroids

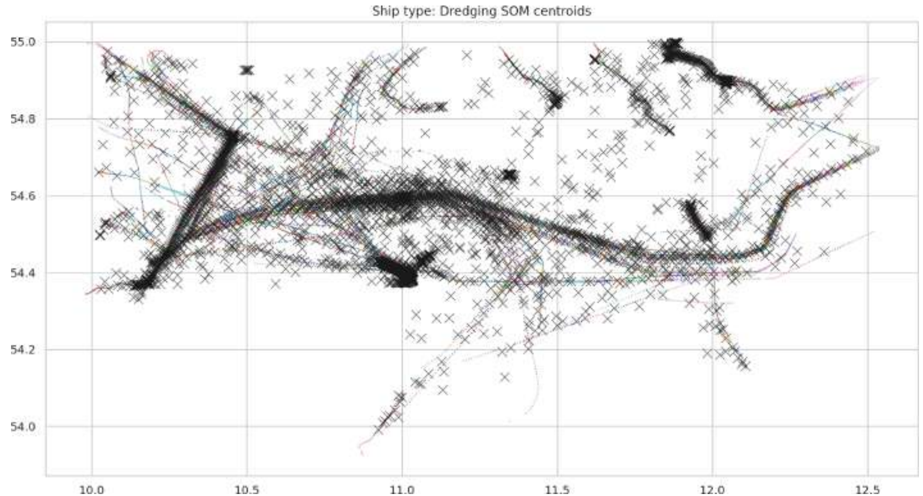




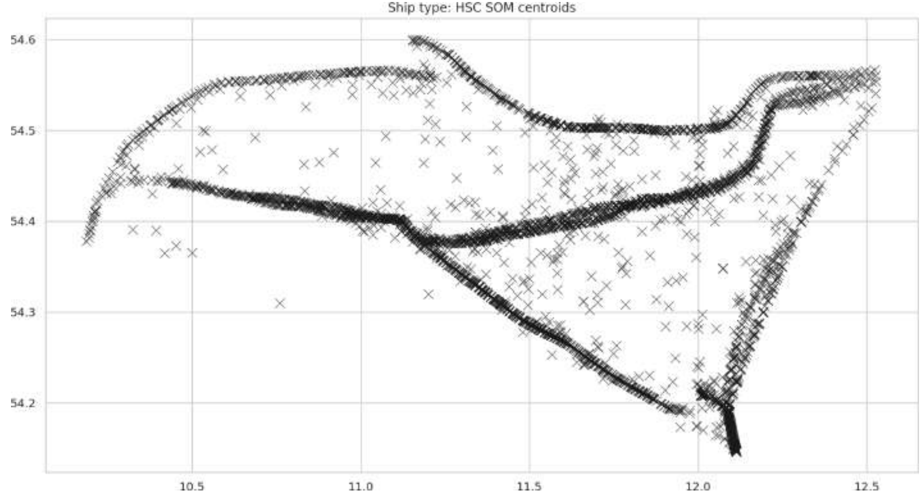
Ship type: Anti-pollution SOM centroids



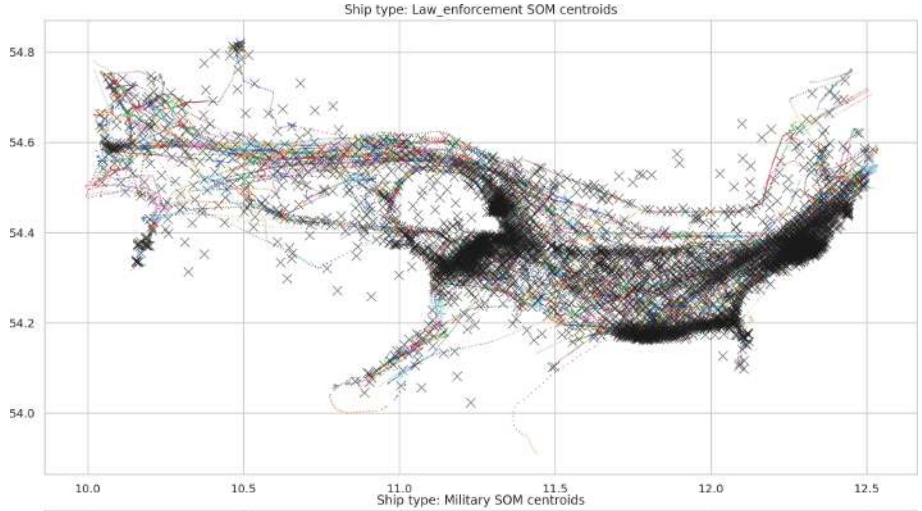
Ship type: Diving SOM centroids

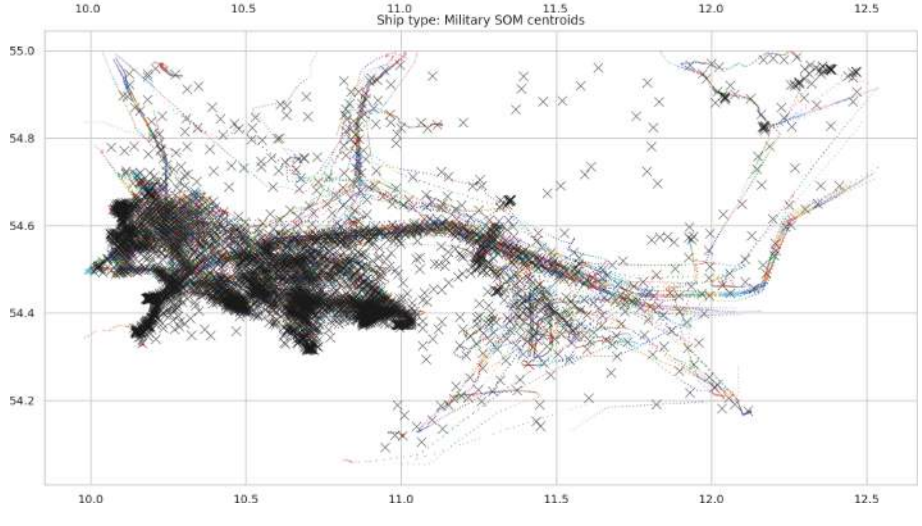


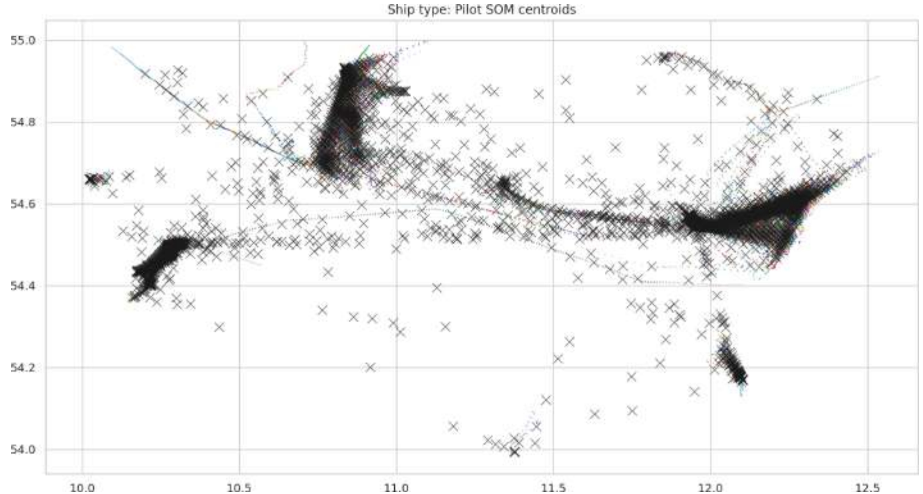
Ship type: Dredging SOM centroids

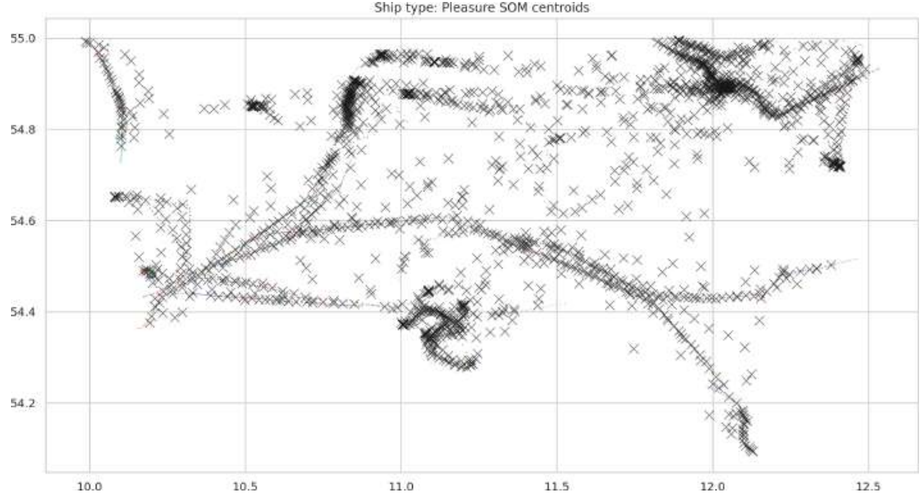


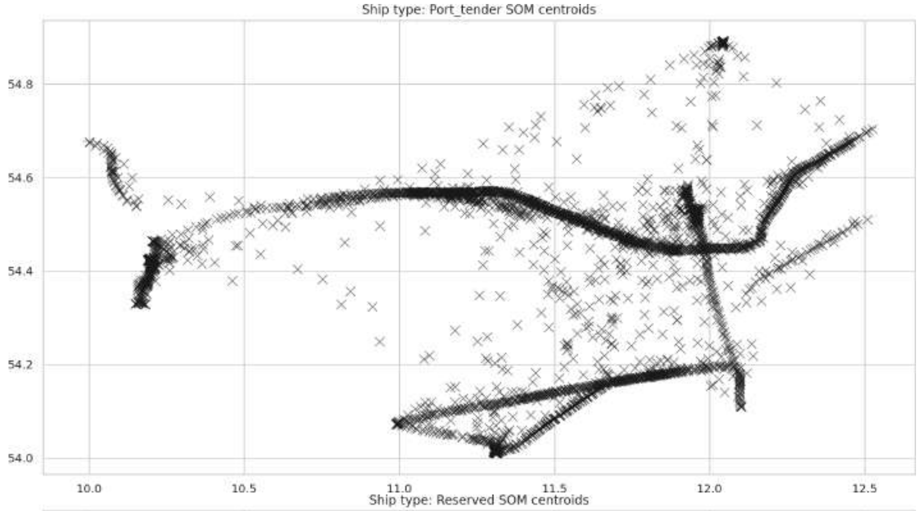
Ship type: HSC SOM centroids

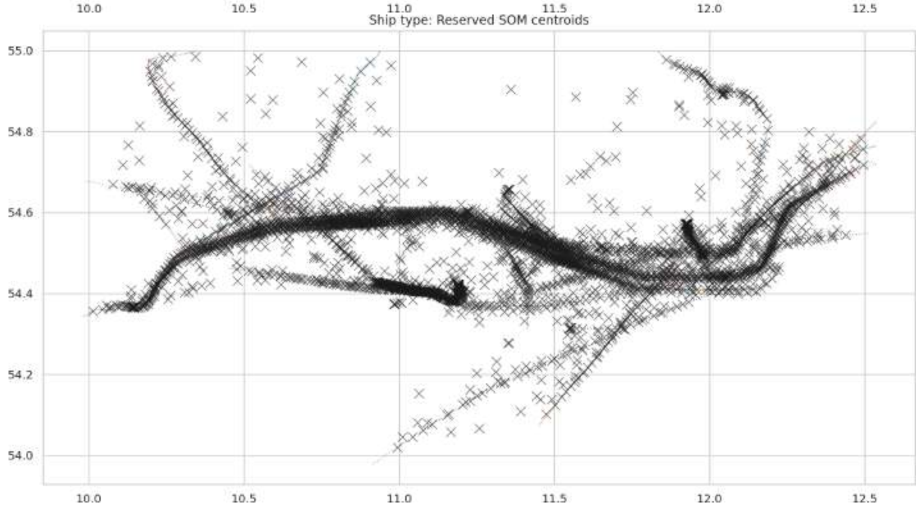

Ship type: Law_enforcement SOM centroids
54.8
54.6
54.4
54.2
54.0
10.0
10.5
11.0
11.5
12.0
12.5

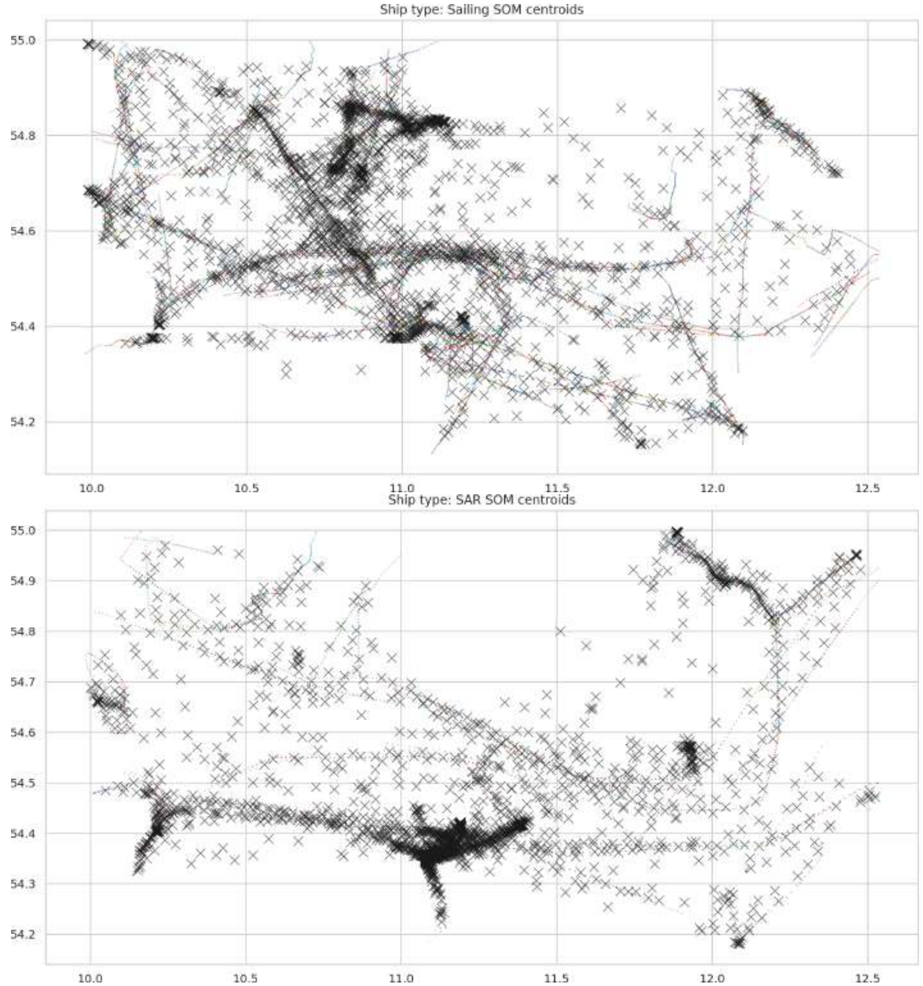

Ship type: Military SOM centroids
55.0
54.8
54.6
54.4
54.2
10.0
10.5
11.0
11.5
12.0
12.5

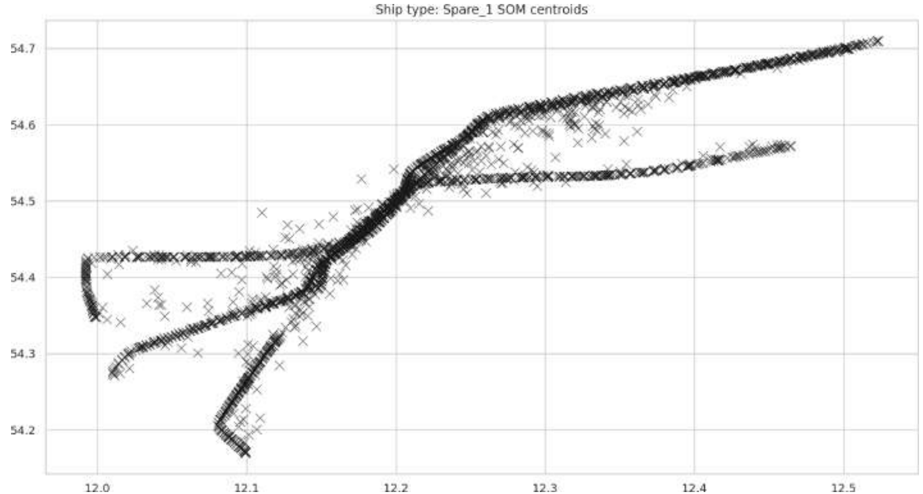

Ship type: Pilot SOM centroids
55.0
54.8
54.6
54.4
54.2
54.0
10.0
10.5
11.0
11.5
12.0
12.5


Ship type: Pleasure SOM centroids
55.0
54.8
54.6
54.4
54.2
10.0
10.5
11.0
11.5
12.0
12.5

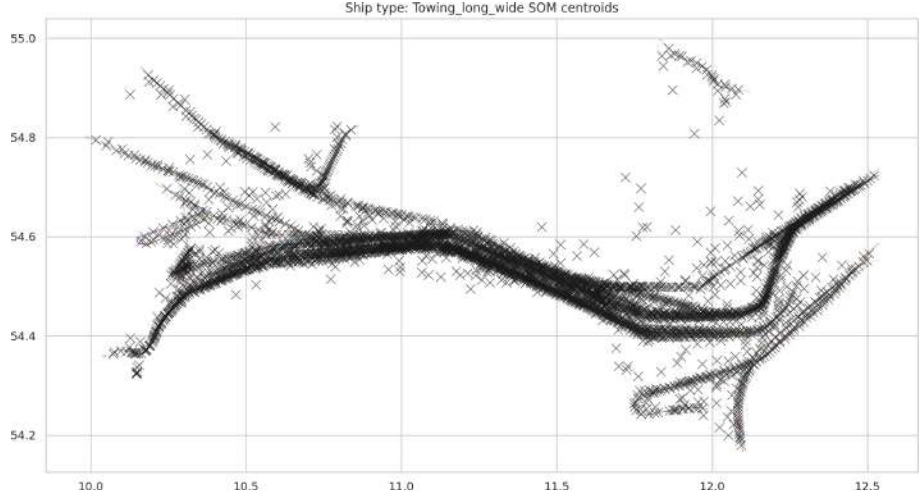

Ship type: Port_tender SOM centroids
54.8
54.6
54.4
54.2
54.0
10.0
10.5
11.0
11.5
12.0
12.5


Ship type: Reserved SOM centroids
55.0
54.8
54.6
54.4
54.2
54.0
10.0
10.5
11.0
11.5
12.0
12.5

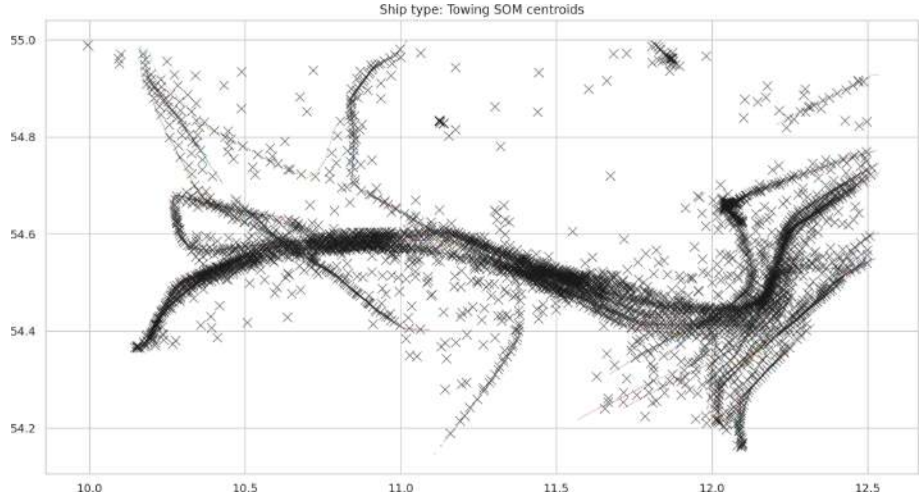

Ship type: Sailing SOM centroids
55.0
54.8
54.6
54.4
54.2
10.0
10.5
11.0
11.5
12.0
12.5
Ship type: SAR SOM centroids
55.0
54.9
54.8
54.7
54.6
54.5
54.4
54.3
54.2
10.0
10.5
11.0
11.5
12.0
12.5

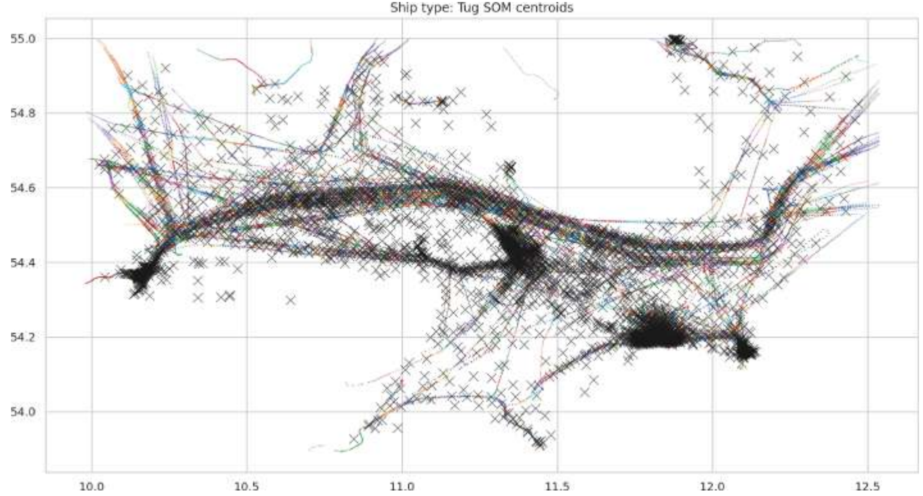


Ship type: Spare_1 SOM centroids



Ship type: Towing_long_wide SOM centroids


Ship type: Towing SOM centroids
55.0
54.8
54.6
54.4
54.2
10.0
10.5
11.0
11.5
12.0
12.5


Ship type: Tug SOM centroids
55.0
54.8
54.6
54.4
54.2
54.0
10.0
10.5
11.0
11.5
12.0
12.5

Julius Venskus

SEMI-SUPERVISED AND UNSUPERVISED MACHINE LEARNING METHODS FOR SEA TRAFFIC ANOMALY DETECTION

DOCTORAL DISSERTATION
Technological Sciences
Informatics Engineering T 007

Editor Liutauras Bartašius

Vilniaus universiteto leidykla
Saulėtekio al. 9, III rūmai, LT-10222 Vilnius
El. p.: info@leidykla.vu.lt, www.leidykla.vu.lt
Tiražas 20 egz.