

Unsupervised marine vessel trajectory prediction using LSTM network and wild bootstrapping techniques

Julius Venskuskas^{a,b} , Povilas Treigys^a, Jurgita Markevičiūtė^{c,1} 

^aInstitute of Data Science and Digital Technologies,
Vilnius University, Akademijos str. 4, Vilnius
julius.venskuskas@mif.vu.lt; povilas.treigys@mif.vu.lt

^bDepartment of Informatics and Statistics, Klaipėda University,
Herkaus Manto str. 84, Klaipėda

^cInstitute of Applied Mathematics, Vilnius University,
Naugarduko str. 24, Vilnius
jurgita.markeviciute@mif.vu.lt

Received: October 1, 2020 / **Revised:** February 17, 2021 / **Published online:** July 1, 2021

Abstract. Increasing intensity in maritime traffic pushes the requirement in better prevention-oriented incident management system. Observed regularities in data could help to predict vessel movement from previous vessels trajectory data and make further movement predictions under specific traffic and weather conditions. However, the task is burden by the fact that the vessels behave differently in different geographical sea regions, sea ports, and their trajectories depends on the vessel type as well. The model must learn spatio-temporal patterns representing vessel trajectories and should capture vessel's position relation to both space and time. The authors of the paper proposes new unsupervised trajectory prediction with prediction regions at arbitrary probabilities using two methods: LSTM prediction region learning and wild bootstrapping. Results depict that both the autoencoder-based and wild bootstrapping region prediction algorithms can predict vessel trajectory and be applied for abnormal marine traffic detection by evaluating obtained prediction region in an unsupervised manner with desired prediction probability.

Keywords: unsupervised traffic anomaly prediction, LSTM prediction region learning, wild bootstrap, autoencoder, joint supervision.

1 Introduction

The maritime logistic industry is a very crucial component of the global trade economy with an expanding volume, traffic intensity, and requirements. In 2019 Q1 to Q3, 2660 million tons gross weight of seaborne goods handled in EU-27 main ports [5]. This is 7 %

¹The author is supported by the Research Council of Lithuania, grant No. S-MIP-17-76.

more in comparison with the same quarters in 2016. Totally, more than 90 % of cargo is transported by sea [27]. Increasing intensity in maritime traffic pushes the requirement in better prevention-oriented incident management system. One of the control technique of this complex management system is the abnormal vessel movement detection. Detection is based on predicting of vessel trajectory by navigational data sequences analysis and search of irregular, illegal, and other anomalous appearances in trajectory/navigational data [9]. The vessel traffic anomaly detection task can be defined as a task of outlier detection, where vessel traffic data are being analyzed as multiple standalone vessels positions/navigational vectors (point-based) or in trajectory-based manner, where vessel's vectors are structured to time series sequences [19]. Automated marine traffic data gathering systems returns huge vessels trajectory/navigational data sets, which are challenging for human-based analysis and traffic anomaly detection [28]. In general, marine traffic is a dynamic system, where vessel's traffic properties change in space and time. Such type of data can be defined as spatio-temporal time series. However, despite advances in prediction of spatio-temporal data with deep neural network, the authors do not propose prediction/confidence interval evaluation that is crucial for marine traffic anomaly detection with this method. Cruz et al. [7] has proposed univariate solution for LSTM prediction interval estimate by joint supervision, but is not sufficient because marine traffic is defined by multivariate coordinates and the proposed method need to be improved. Recently published works take advantages of extended LSTM (Long Short Term Memory) neural networks to learn spatio-temporal dependencies (see [12, 15, 20]).

This paper investigates and proposes a method based on LSTM autoencoder [18] to predict vessel trajectory and evaluate prediction region. The paper extends previous authors' investigations in field of marine traffic anomaly detection [21–26]. The paper is structured as follows: Section 2 describes proposed methodology for abnormal marine traffic detection. Section 2.1 describes data preparation before trajectory patterns learning. Section 2.2 depicts LSTM network architecture used for vessel movement prediction. Next, Section 2.2 describes a method for LSTM prediction region evaluation training, custom loss functions. Then Section 2.2 description of wild bootstrapping is presented with the view to be able to compare predictions. Section 3 reveals the experimental investigation of proposed methods, depicts the parameters and other settings that were used for model results validation. Finally, Section 4 concludes the paper.

2 Experiment setup

2.1 Data set preparation

Data structuring. The source of the vessel traffic data is AIS. Navigational information includes data about vessel current geographical location coordinates in WGS84 geodetic system, heading, course over ground (COG), speed over ground (SOG), maritime mobile service identity (MMSI) for unique vessel identification, navigational status, vessel type, destination port, vessel length/width, draught, callsign, name, cargo type, and estimated time of arrival (ETA).

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

$$\begin{aligned} X &= \{x_1, x_2, \dots, x_i, \dots, x_{o-1}, x_o\}, \\ x_i &= [x^{(1)}, x^{(2)}, \dots, x^{(j)}, \dots, x^{(f-1)}, x^{(f)}], \end{aligned} \quad (1)$$

where $x_i, i = 1, 2, \dots, o$, is a single vessel navigational data record that consist $x^{(j)}, j = 1, 2, \dots, f$, parameters as: vessel unique identifier MMSI, “latitude”, “longitude”, SOG, COG, ship type, timestamp of the data being received. Whole data set contains o number of records, where each record is set of f parameters of vessel navigational vector.

Vessels send navigational data periodically, thus received data instance is stored in ordered manner by the timestamp as the data was received. This way structured raw data is complicated to work with distinctive vessels multiple navigational data that forms vessel’s sea path over time. In order to develop a model that recognizes the unusual behavior of vessel’s traffic, we need to train model to predict single vessel path. To achieve that, authors restructure data per unique vessel, vessel’s navigational data is grouped by unique vessel identifier MMSI and ordered by the timestamp:

$$S(X) = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_v \end{bmatrix} = \begin{bmatrix} \{x_{(1,1)}, x_{(1,2)}, \dots, x_{(1,b_1)}\} \\ \{x_{(2,1)}, x_{(2,2)}, \dots, x_{(2,b_2)}\} \\ \vdots \\ \{x_{(v,1)}, x_{(v,2)}, \dots, x_{(v,b_v)}\} \end{bmatrix}, \quad (2)$$

where $S(X)$ is a data restructuring function that restructures the navigational data vectors of each ship into matrix rows according to the ship’s MMSI parameter $x^{(\text{MMSI})}$. Each row of the matrix consists of a set of navigational data vectors s_1, s_2, \dots, s_v of an individual vessel, where v is the number of distinct vessels. The sets navigational data of each vessel are compound of the navigational vectors $x_{(v,b_v)}$, and b_v is the number of navigational vectors of a specific vessel. The vectors of each vessel in the set are sorted by the data acquisition time parameter when $x_{(v,b_v-1)}^{(\text{time})} < x_{(v,b_v)}^{(\text{time})}$. Vectors received earlier in time are marked with lower index number.

Data cleaning. AIS systems interconnect many participants such as vessels, vessel traffic services (VTS), receiving base stations, and collection databases. The Danish maritime agency has collected in territorial waters during year 2019 more than 1 TB and 10^9 vessel navigational vectors [1]. 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 collision with other vessel data transmission.

To overcome these problems, the gathered vessel data from the AIS needs to be cleaned. All vectors that are outside of analyzed marine area has to be dropped (the selected areas of the sea traffic will be depicted in Section 3.2), all duplicated data and anchored vessel navigational vectors are removed. Short sequences of vessel vectors (paths) shorter as predefined training sequence length n' are removed as well. The cleaning of vessel path gaps are depicted in the paragraph *Splitting to sequences of vessels*

navigational vectors below, ensuring that the generated sequence contains successive vectors between which the time interval of acquisition $\Delta t = x_{(v,b_v)}^{(\text{time})} - x_{(v,b_v-1)}^{(\text{time})}$ is lower than predefined threshold $T_{\Delta t}$, otherwise, sequences are discarded. The threshold is described in process of method tuning in Section 3 of this paper.

Filling missing values. The collected vessel traffic data may have missing features values in navigational vectors. Depending on the type of missing feature, a different strategy for filling in the missing values may be chosen. Below, authors present the variety missing feature value types and further depict the strategies that were applied to solve the missing value issue.

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

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

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

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

Strongly correlated missing values of dynamic feature. Features that strongly correlate with other features within the same vessel sent data (properties of vessel are very similar, or vessel's AIS transceiver don't have information from all on-board sensors). As an example, this could be heading, which strongly correlates with course over ground (COG). Also, on some vessels, for unknown technical reasons, Heading field is not sent or is replaced with the values of COG.

Weakly correlated missing values of dynamic feature. Features that has weak or very weak correlation with other features. As example, it can be vessel rate of turn (ROT), estimated time of arrival at the port of destination, etc.

For each group of missing value types, a different value correction strategy is applied.

Partially missing static features values – missing values are filled in by searching actual value withing same vessels' navigational data. After it is found, the rest of vectors are filled with that value, otherwise, is treated as completely missing static features.

Completely missing static features values – depends on type and property of feature and results in two techniques: to fill missing values with predictive model that 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, 0.95 threshold were used. Such approach is proposed in previous work [22]; either discard the entire attribute from all data for all vessels, or if the number of missing values low and mainly relates to small amount of vessel, then we drop particular vessels from 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 are missing of such feature, then all navigational data of particular vessels that miss values is dropped from data set.

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

The minimum set of features must be maintain such as longitude, latitude, Speed over ground (SOG), Course over Ground (COG), Wind direction, Wind speed, Wave direction, and Height.

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

Feature engineering. The lack of constant data intervals results in variation of sailed distance at same speed while analysing a vessel's subsequent vectors. In order to solve the issue, a new differential features are introduced. First, a new feature introduced is a time difference between sequential vectors' timestamps expressed by

$$x_{(v,b_v)}^{(\Delta t)} = x_{(v,b_v)}^{(\text{time})} - x_{(v,b_v-1)}^{(\text{time})}, \quad (3)$$

where $x_{(v,b_v)}^{(\Delta t)}$ is new feature, $x_{(v,b_v)}^{(\text{time})}$ is particular navigational data vector's time of same vessel, and $x_{(v,b_v-1)}^{(\text{time})}$ is previous data vector receive time. v is particular vessel's data set number, b_v is vessels position vectors number in sequence as in formula (2). Two more introduced features are extracted to express vessel movement differential in time for latitude and longitude as follows:

$$x_{(v,b_v)}^{(\delta \text{Lat})} = \frac{x_{(v,b_v)}^{(\text{Lat})} - x_{(v,b_v-1)}^{(\text{Lat})}}{x_{(v,b_v)}^{(\Delta t)}}, \quad x_{(v,b_v)}^{(\delta \text{Lon})} = \frac{x_{(v,b_v)}^{(\text{Lon})} - x_{(v,b_v-1)}^{(\text{Lon})}}{x_{(v,b_v)}^{(\Delta t)}}. \quad (4)$$

Here $x_{(v,b_v)}^{(\delta \text{Lat})}$ and $x_{(v,b_v)}^{(\delta \text{Lon})}$ are newly constructed features based on latitude and longitude differences in subsequent vectors $x_{(v,b_v)}^{(\text{Lon})}$, $x_{(v,b_v-1)}^{(\text{Lon})}$ of same vessel.

In addition to that, earlier works has shown that meteorological data has significant influence to a marine traffic models [22]. This data includes information about wind direction, wind strength, swell direction, swell height, swell period, day/night, tide level.

Above mentioned features are as well artificially added to each vessel data vector that were registered by AIS system. Meteorological data is taken periodically from the European Centre for Medium-Range Weather Forecasts (ECMWF) grid. ECMWF provides data in certain geographical interpolated resolution. Vessel position accuracy is much higher than meteorological data grid, thus the assignment of particular grid point to vessel navigational vector has to be accomplished. It can be several meteorological data providers. Used in this research described in Section 3. Meteorological data is assigned to a navigational vectors by using the algorithm of the nearest neighbour in location-wise and time-wise manner. At first, distance to all meteorological locations is calculated by using haversine formula [4] to WGS84 geodetic system coordinates of vessel and meteorological locations. Closest meteorological location is assigned based on calculated distances, and then closest in time forecast is picked from all forecasts for this location. This meteorological data is assigned to vessel position vector.

Splitting to sequences of vessels navigational vectors. In upcoming section, for vessel position prediction, the algorithm of artificial deep neural network (DNN) is described. In order to obtain 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 structured by $S(X)$ function (2). Window processes each vessel's data set s_v separately and slices $x_{(v,1)}, x_{(v,2)}, \dots, x_{(v,b_v)}$ set in to sequences in length of $\tilde{n} + n'$, where \tilde{n} is DNN output sequence (prediction) length, and n' is input (for prediction) sequence length. The window slide according to 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 \tilde{n} is written into output matrix. The obtained matrices can be defined by expressions

$$(\chi, Y) = \left(\begin{bmatrix} x_{(1,0+1)} & x_{(1,0+2)} & \dots & x_{(1,0+n'-1)} & x_{(1,0+n')} \\ x_{(2,1+1)} & x_{(2,1+2)} & & x_{(2,1+n'-1)} & x_{(2,1+n')} \\ \vdots & & \ddots & & \vdots \\ x_{(v,N+1)} & x_{(v,N+2)} & \dots & x_{(v,N+n'-1)} & x_{(v,N+n')} \end{bmatrix}, \begin{bmatrix} x_{(1,n'+0+1)} & x_{(1,n'+0+2)} & \dots & x_{(1,n'+0+\tilde{n}-1)} & x_{(1,n'+0+\tilde{n})} \\ x_{(2,n'+1+1)} & x_{(2,n'+1+2)} & & x_{(2,n'+1+\tilde{n}-1)} & x_{(2,n'+1+\tilde{n})} \\ \vdots & & \ddots & & \vdots \\ x_{(v,n'+N+1)} & x_{(v,n'+N+2)} & \dots & x_{(v,n'+N+\tilde{n}-1)} & x_{(v,n'+N+\tilde{n})} \end{bmatrix} \right), \quad (5)$$

where χ is model's input, Y is model's output, N is number of vessel navigational vectors sequences, n' – single sequence length for input, \tilde{n} – navigational vectors sequence length for output, χ and Y are matrices of vessels' navigational vector sequences formed by sliding window process while assigning a vector $x_{(v,b)}$ from current window position to a sequences matrices (χ, Y) . Obtained matrices for further model creation are split into subsets that are used for model training, validation, and testing by random row selection at ratio 60 : 20 : 20 percents accordingly.

2.2 Method for trajectory prediction

For vessel trajectory prediction, the paper applies the deep neural network. 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 defined limit, the vessel's expected location is considered as normal, otherwise, abnormal. The more detailed scheme of the prediction evaluation is presented in Sections 3.3 and 3.4.

Long short term memory neural network. Fully connected dense artificial neural networks does not assure history retrospective. Recurrent neural networks (RNN) was proposed to overcome this challenge and to overcome practical challenges vanishing gradients when longer sequences of input data is used, the paper proposes to use long short term memory (LSTM) network [13] that performs significantly better in other applications: speech recognition [3], handwriting recognition [10], reinforcement learning [11], 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 nonlinear patterns. LSTM network architecture represents interconnected cells. LSTM cells transmit cell state $c(t)$ 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 $y(t)$. From previous cell $h(t-1)$ this cell receives hidden state, $c(t-1)$ cell state, and $x(t)$ cell input. Then the cell computes what information should be kept for further calculations and what has to be forgotten [13]. By interconnecting such cells the architecture of LSTM autoencoder can be obtained.

LSTM autoencoder. An autoencoder is a type of artificial neural network used to learn efficient data encoding in an unsupervised manner [14]. A typical autoencoder is compound of three parts: encoder, latent vector, and decoder. During a training, encoder as well as 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 autoencoder is that main blocks of the network architectures are LSTM cells. This paper investigates the LSTM autoencoder by providing the sequences of vessel's navigational vector data. Encoder compresses input data X equation (5) to latent space, and decoder forecasts sequence of next vessel positions Y with limited set of spatial features, like latitude and longitude. In this paper, authors use a multivariate multistep LSTM autoencoder shown in Fig. 1. The main parts of a proposed LSTM autoencoder 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 (5) and returns the predicted/reconstructed output sequences \hat{Y} with limited number of feature set, i.e. latitude and longitude. Encoder and decoder parts are compound of LSTM cells that are interconnected sequentially inside particular block and are parallel between on-top-stacked layers. A reconstruction (prediction) error of autoencoder is obtained by

$$e_{l,i,j} = Y_{i,j} - \hat{Y}_{l,i,j}, \quad l = \{\text{upper, crisp, lower}\},$$

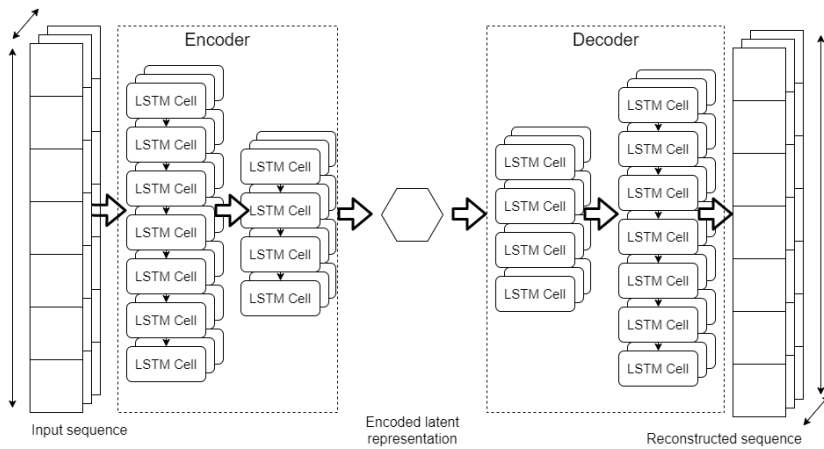


Figure 1. Architecture of LSTM autoencoder.

$i = 1, 2, \dots, \tilde{n}$, $j = 1, 2, \dots, f'$, where $e_{l,i,j}$ is the error of a reconstruction single navigational vector's feature, $Y_{i,j}$ – true value of a single navigational vector feature, $\hat{Y}_{l,i,j}$ – output estimated value by autoencoder of a single navigational vector feature, l is model type: crisp, lower, upper (described in Section 2.2 below), i – position in sequence of predicted navigational vector, \tilde{n} – output sequence length, j – j th navigational vector's feature, f' – number of output features, Y and \hat{Y} are navigational vectors structured by expression (5). For crisp version of network, a classical mean squared error (MSE) loss function is used:

$$L_s^l = \frac{1}{N\tilde{n}f'} \sum_{g=1}^N \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{f'} e_{l,g,i,j}^2, \quad l = \{\text{upper, crisp, lower}\}, \quad (6)$$

where L_s^l is the loss function for l -type of model, N – number of training sequences in the training data set.

LSTM prediction region learning. In order to determine the abnormal vessel traffic, authors investigate the ability to check real vessels navigational vectors against predicted ones by the model in two-dimensional space. The assumption is as follows: if vessel's true position vector lays outside of prediction region (multivariate case of prediction interval), it is interpreted as abnormal vessel movement, and all vessel traffic vectors that results inside the prediction region are interpreted as normal vessel movement. The authors investigate and compare two methods for prediction region calculation. The first method, based on LSTM autoencoder training, is described further, and the second, based on a wild bootstrapping method, is described in the subsequent Section 2.2.

LSTM autoencoder in typical configuration calculates only most like-hood forecast (crisp). In order to determine prediction regions, a method proposed by Cruz at al. [7] is used with modification to support multivariate and multistep-type LSTM networks. Prediction region is composed of an upper and a lower bounds in which the prediction/

reconstruction output is found with certain probability α [16]. The region is learned by training two LSTM autoencoders with combined classical MSE loss function (6) with the second metric of region loss function as presented in [6]. The specific loss function for upper and lower bounds is defined as follows:

$$L_{\ell}^{\text{upper}} = \frac{1}{N\tilde{n}f'} \sum_{g=1}^N \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{f'} (\text{ReLU}(e_{\text{upper},g,i,j}))^2, \quad (7)$$

$$L_{\ell}^{\text{lower}} = \frac{1}{N\tilde{n}f'} \sum_{g=1}^N \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{f'} (\text{ReLU}(-e_{\text{lower},g,i,j}))^2, \quad (8)$$

where L_{ℓ}^{upper} and L_{ℓ}^{lower} is specific loss function for upper and lower bounds, respectively, ReLU is rectified linear unit function defined by

$$\text{ReLU}(x) = \begin{cases} 0 & \text{for } x < 0, \\ x & \text{for } x \geq 0. \end{cases} \quad (9)$$

As presented in paper [6], data points $Y_{i,j}$ larger than L_{ℓ}^{upper} apply a cost equivalent to the squared difference between the real data point and its upper bound prediction/reconstruction in accordance with Eq. (7). Likewise, data points $Y_{i,j}$ lower than L_{ℓ}^{lower} is penalized as defined in Eq. (8). Data points $Y_{i,j}$ that are in prediction region (below upper and above lower bounds) has no cost with a help with ReLU function (9).

In combination of the upper and lower loss functions, a higher loss value is applied for $Y_{i,j}$ points that are outside of the prediction region. This 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 (6), and the region loss functions (7), (8) – for upper and lower bounds, respectively [6, 7]:

$$L_{\text{total}}^{\text{upper}} = L_s^{\text{upper}} + \lambda L_{\ell}^{\text{upper}}, \quad (10)$$

$$L_{\text{total}}^{\text{lower}} = L_s^{\text{lower}} + \lambda L_{\ell}^{\text{lower}}, \quad (11)$$

where $L_{\text{total}}^{\text{upper}}$ is overall upper loss function, $L_{\text{total}}^{\text{lower}}$ – overall lower loss function, λ – a tuneable parameter that represents the relative importance of proposed classical/common and region loss functions [7]. The crisp model's output is learned by using only a MSE loss function (6):

$$L_{\text{total}}^{\text{crisp}} = L_s^{\text{crisp}}, \quad (12)$$

where $L_{\text{total}}^{\text{crisp}}$ is loss function for crisp model.

With these loss functions, the minimization of prediction region area is achieved. If functions is not applied, the prediction region loss functions (L_{ℓ}) 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. (10) and (11).

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 quantify the number of measured values that fall within the region defined by the model [7] and modified to support multivariate features and multistep predictions:

$$\text{PICP} = \frac{1}{N\tilde{n}f'} \sum_{g=1}^N \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{f'} (\delta_{g,i,j}), \quad (13)$$

$$\delta_{g,i,j} = \begin{cases} 1 & \text{if } Y_{g,i,j} \in [\hat{Y}_{g,i,j}^{\text{lower}}, \hat{Y}_{g,i,j}^{\text{upper}}], \\ 0 & \text{otherwise.} \end{cases}$$

The second metric is prediction region normalized average width (PINAW) that is used to measure the area of the region [7] also modified for multistep and multivariate features:

$$\text{PINAW} = \frac{1}{N\tilde{n}f'R} \sum_{g=1}^N \sum_{i=1}^{\tilde{n}} \sum_{j=1}^{f'} (\hat{Y}_{g,i,j}^{\text{upper}} - \hat{Y}_{g,i,j}^{\text{lower}}),$$

where R is the distance between the maximum and minimum values measurements $\max(\hat{Y}_{g,i,j}^{\text{upper}} - \hat{Y}_{g,i,j}^{\text{lower}})$ in the data set [6, 7].

The algorithm to train the network iterative way is shown [6]. In this approach, the λ parameter is increased iteratively to force a wider region area in each of the iteration as the coverage probability increases. In each iteration, the PICP is estimated by Eq. (13) [6]. When the desired coverage probability α has been achieved, the algorithm stops the λ parameter increment. Few more iterations are calculated using fixed λ parameter in order to compensate the random initialization of the initial algorithm weights [6].

Wild bootstrap prediction region. 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 resamples the initial data to produce more data samples that could be used in repetitive experiment. However, instead of generating bootstrap samples that consist of resampling the original data or residuals, the wild bootstraps combine the data with random variables drawn from a known distribution to form a bootstrap sample. Bootstrap usage in this paper can be summarised by applied further steps:

- (i) Prepare data as described in Section 2.1.
- (ii) Calculate data set's variance for every feature type.
- (iii) Generate multivariate normal random variables while keeping the same dimension, the mean equal to zero, and the variance the same as that of the input data.
- (iv) Element-wise sum the initial data set with the newly generated, i.e. add noise to the data with mean and variance calculated from initial data set.
- (v) Scale resulting data for better LSTM training results into interval $[0, 1]$ while keeping each feature scaling factors for predicted data reconstruction purpose.
- (vi) Train LSTM autoencoder network.

- (vii) Calculate LSTM network predictions a -step ahead, $a \in \{1, 2, \dots, m\}$ (authors chose $m = 50$).
- (viii) Restore prediction scaling, i.e. upscale predicted values according to the saved feature's scaling parameters described two steps above.
- (ix) Repeat steps (iii)–(viii) k -times (authors repeated experiment 100 times).

After the application of the proposed above scheme, 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 [2])

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

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

3 Experiments

With the view to validate and test prediction approaches being presented, the experiments were performed with real marine traffic data. The data was prepared as described above. Further in this section, the experiment setup is being described together with the obtained results. The LSTM networks were trained, and wild bootstrapping technique was applied, predictions regions calculated.

3.1 Description of the data sets

AIS data set. The AIS data set for this research are obtained from Danish Maritime Authority [1] and contains historical AIS maritime vessel traffic in Danish waters data from 2006 year to 2020 year. Records of this data sets contain single vessel navigational vector and presented by Eq. (1). Single record has such data fields: Timestamp – timestamp from the AIS base station; Type of mobile describes what type of target this message is received from; MMSI – MMSI number of vessel; Latitude – latitude of message report (e.g. 57,8794); Longitude – longitude of message report (e.g. 17,9125); Navigational status – navigational status from AIS message if available, e.g.: “Engaged in fishing”, “Under way using engine”, and etc.; ROT – rot of turn from AIS message if available; SOG – speed over ground from AIS message if available; COG – course over ground from AIS message if available; Heading – heading from AIS message if available; IMO – vessel identifier provided by International Maritime Organization; Callsign – callsign of the vessel; Name – name of the vessel; Ship type – describes the AIS ship type of this vessel; Cargo type – Type of cargo from the AIS message; Width – width of the vessel; Length – length of the vessel; Type of position fixing device – type of positional fixing

device from the AIS message; Draught – draught field from AIS message; Destination – destination from AIS message; ETA – estimated time of arrival if available; Data source type – data source type, e.g. AIS.

Meteorological data set. Meteorological data set obtained from World Weather Online service API [17] (application programming interface) 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, tide level. Meteorological data is provided periodically in 3 hours periods. Data collected in period from 2019 November 1 to 2020 June 31.

3.2 Data preprocessing

For method validation, the data time interval is taken from 2019 November 1 till 2019 November 30, and geographical region – between 54.2620 – 54.8292° of latitude and 10.6897 – 12.9694° of longitude. The region was chosen in order to have intense marine traffic and variety of different vessel types. Both vessel AIS navigational and meteorological data is registered and used in experiment from time period and geographical region described above with no duplicates.

In earlier research performed by the authors, was found that different vessels types introduce different traffic patterns [26]. Each vessel type has unique traffic patterns. By taking into account that each vessel acts according different pattern and the navigational data sets representing each vessel type are unbalanced, authors decided to investigate separate models for each vessel type. Authors selected “Cargo” vessel type because of largest data amount compared to other vessel types. Data was stored in Ψ defined by (1) data structure, containing number of navigational vectors r for that time period is 7331756, and number of initial parameters f was 22. Then data set restructured per unique vessel identifier MMSI and ordered by the timestamp by structure defined in (1). The preparation of the data resulted in 1144 distinct vessels v . The predefined calibrated parameter $\Delta T_{\text{interval}}$ is chosen 2 minutes. This parameter can be calibrated based on awareness requirements of Vessel Traffic Service (VTS). This research is based on assumption that anomaly detection will performed in middle range of vessel activity, that is, average 30 % of activity in region of interest. In average, the vessels pass this “Fehmarnbelt” region in 5–10 hours. Minimum time to detect trajectory anomaly required is 1.6–2.4 hours. If $n' = 50$, then 2 minutes falls in this range. During down-sampling, the nearest neighbour algorithm was used to select down sampled values.

As described in Section 2.1, the vessels with very few navigational vectors emerge because of noise introduced by the nature of AIS, and those vessels that just entered region of interest. All vessels with vectors that are shorter than $n' + \tilde{n}$ and for this case is chosen $n' = 50$, $\tilde{n} = 50$, are removed. This made because we are not interested in vessel that are leaving geographical region of interest. Another issue related with the data is speed over the ground. It is observed that some data contains vectors with 0 knots speed. As we are interested in only moving vessel new position in space predictions, vectors having SOG equal to 0 are removed from the data. Only moored or anchored vessels has 0 value

of SOG. Since this paper analyzes anomaly of moving vessel behaviour, the moored or anchored is removed from data set to balance data set accordingly.

After data cleaning, the filling missing values and removing unnecessary features takes part. Fields “Type of mobile”, “Type of position fixing device”, and “Data source type” are removed because contains data only related to data transmission, but to vessel traffic. Fields “IMO” and “Callsign” are removed as they represents unique identifier of the vessel same as MMSI. Fields “ROT”, “Cargo type”, and “Destination” is removed because majority of vessels are missing these data. Then all missing values are filled based on description in Section’s 2.1 *Filling missing values* paragraph. Additional features were engineered by applying Eqs. (3), (4) to express vessel movement differential in time for latitude and longitude, then each vessel navigational vector enriched with meteorological information as described in Section 2.1.

Splitting to sequences of vessels navigational vectors performed by window approach described in Section 2.1 under paragraph *Splitting to sequences of vessels navigational vectors*. Authors took data structure as described by Eq. (2) and spited the data with moving window by applying step $\mu = 50$ to set of sequences X and Y in $n' = 50$, $\tilde{n} = 50$. During splitting, algorithm follows a few additional rules: firstly, one sequence must contain only subsequential vectors of same vessel; secondly, if time difference of two subsequent vectors $\Delta t = \psi_{v,b_v}^{\text{time}} - \psi_{v,b_v-1}^{\text{time}}$ is larger than predefined threshold $T_{\Delta t} = 10$ min, the sequence is discarded. This step allows to filter sequences with region of interest crossing vessels. After data split, total number of sequences obtained is $N_{\text{total}} = 26214$ that further was randomly shuffled and divided to training/validation and testing data subsets while keeping ratio 80 : 20 ($N = 20971$ to $N_{\text{test}} = 5243$) accordingly. The test data set is kept untouched until final evaluation of model. The training/validation data set is split randomly before each training epoch in similar ratio 80 : 20 to training/validation set ($N_{\text{train}} = 16777$, $N_{\text{validation}} = 4194$).

3.3 Prediction region estimation

LSTM autoencoder network is applied as that described in Section 2.2. The input shape of network set as input sequence length $n' = 50$, number of features $f = 18$, and batch size of 512. The first LSTM layer has 128 units and forms 50×128 sequential output that connected with the second LSTM layer with 16 units and forms not sequential output. The encoded latent representation vector has 16 units. Third LSTM layer has 16 units and last has 128 units. Output layer size is being used two-dimensional $\tilde{n} = 50$ and $f' = 2$. For the network output, only spatial features latitude and longitude are used. For all LSTM cells, TANH activation function were applied. To ensure cross validation and regularization before each epoch an initial training/validation, data set is randomly divided into training and validation data sets with ratio 80 : 20 percents of training/validation data. Before each epoch, the data set instances are reshuffled and samples assigned to training and validation nonoverlapping data sets. 300 epoch are set for network training. As an optimiser, the Adam algorithm [8] is used with calibrated parameters: learning rate $\alpha_{\text{Adam}} = 0.001$, exponential decay rate for the first moment $\beta_1 = 0.9$, exponential decay rate for the second-moment $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. LSTM networks were configured with different loss

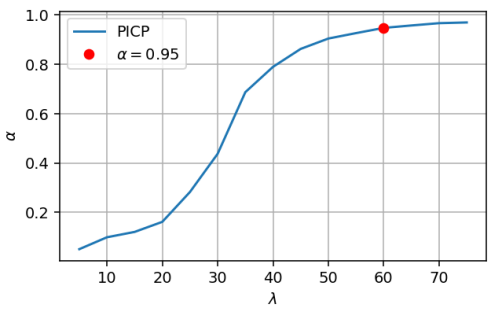


Figure 2. PICP change against λ value.

functions: crisp network has configured to use loss function (12); lower and upper bounds LSTM networks configured by loss function (11) and (10) accordingly. All the models were trained by the algorithm described in Section 2.2, and results are shown in Table 1. Initial value of tuneable parameter $\lambda = 10$ is chosen experimentally and were increased in each step by $\Delta\lambda = 5$ while reached $\text{PICP} = \alpha = 0.95$. Figure 2 depicts change of PICP while training networks at different value of λ . When PICP reaches desired value $\alpha = 0.95$, the trained model for each specific network with that particular λ value are chosen for further investigation. It is worth to note that training upper and lower models start with slow progress and only after 60th epoch loss starts to drop significantly. After all three models types (crisp, lower, upper) were trained, prediction part take place. In order to predict and evaluate traffic anomaly, a timed sequence of navigational vectors are taken for that vessel. This sequence prepared as described in Section 2.1. The first $n' = 50$ prepared vectors forms input X of the models. Lower and upper models produce prognoses \hat{Y}^{upper} and \hat{Y}^{lower} . A vessel movement considered as normal when true value Y of is between $\hat{Y}^{\text{upper}} > Y > \hat{Y}^{\text{lower}}$, and otherwise, considered as abnormal. As the alternative to the LSTM prediction region learning method, the wild bootstrapping method was tested as well. Method was described in Section 2.2. The steps were repeated $k = 100$ times.

3.4 Results

Trained models were tested on test data set. In Fig. 3, there is shown randomly selected cases of normal vessel trajectory. The figure depicts vessel trajectory in longitude and latitude plane. Light blue color shows actual marine traffic (10000 trajectories) in the area. Green circles 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 frames 25 and 50, respectively. Red star represents crisp model prediction $\hat{Y}_{25}^{\text{crisp}}, \hat{Y}_{50}^{\text{crisp}}$ at time frames 25 and 50. Green star shows prediction $\hat{Y}_{25}^{\text{upper}}, \hat{Y}_{50}^{\text{upper}}$ of upper bound model at time steps 25, 52, and blue star marks prediction $\hat{Y}_{25}^{\text{lower}}, \hat{Y}_{50}^{\text{lower}}$ obtained by lower bound model at time steps 25, 52. Red dashed lines depict boundaries of prediction region for particular movement forecast. Blue ellipse bounds prediction region calculated by wild bootstrap method. Trained models were tested on test data set. Figure 4 shows randomly selected cases of normal vessel trajectory.

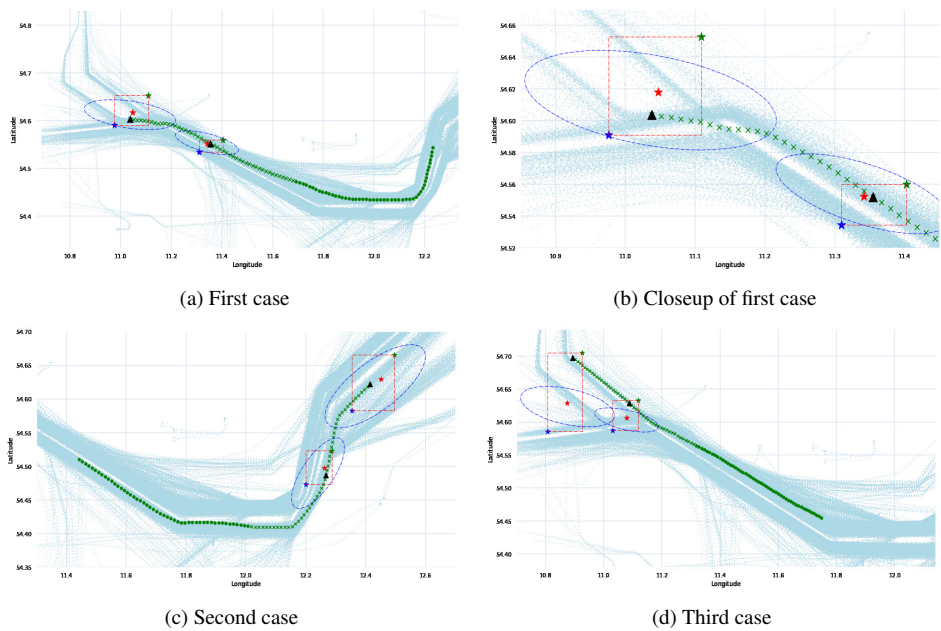


Figure 3. Random cases of *normal* vessel traffic.

The figure depicts vessel trajectory in longitude and latitude plane. Light blue color shows actual marine traffic (10000 trajectories) in the area. Green circles 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 frames 25 and 50, respectively. Red star represents crisp model prediction $\hat{Y}_{25}^{crisp}, \hat{Y}_{50}^{crisp}$ at time frames 25 and 50. Green star shows prediction $\hat{Y}_{25}^{upper}, \hat{Y}_{50}^{upper}$ of upper bound model at time steps 25, 52 and blue star marks prediction $\hat{Y}_{25}^{lower}, \hat{Y}_{50}^{lower}$ obtained by lower bound model at time steps 25, 52. Red dashed lines depict boundaries of prediction region for particular movement forecast. Blue ellipse bounds prediction region calculated by wild bootstrap method.

Figure 3 shows a few cases of vessel normal movement trajectories that passes were passed into model and visualised in two dimensional latitude/longitude planes. The model was only introduced to a sequence of first 50 vessel navigational vectors, that are marked as green circles (seen in Figs. 3(a), 3(d), 3(d)). Green crosses marks subsequential vessel positions Y , which were unknown for the model. Model predicts upper \hat{Y}_{25}^{upper} and lower bounds \hat{Y}_{25}^{lower} shown with green, blue stars and red dotted rectangle. True value lay inside the rectangle and the vessel trajectory should be considered as normal according to LSTM prediction model. If true value lay inside blue ellipse, then the vessel trajectory considered as normal by wild bootstrap method. Both black triangles is in red rectangle, but only one is inside blue ellipse. This situation illustrates the case when the LSTM prediction region method indicates traffic as normal while bootstrapping method as abnormal. It is observed that the narrow marine traffic area is the smaller region models learns and

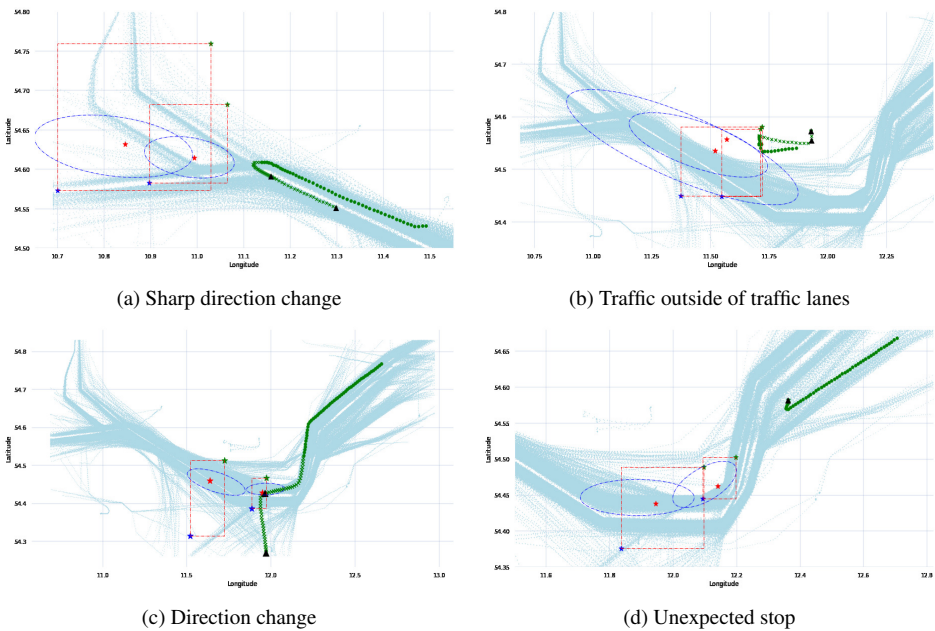


Figure 4. Different types of *abnormal* vessel traffic.

wise versa. This is seen in Fig. 3(d), where the first prediction region is smaller because it is on vessel's routes junction. In opposite, if vessels routes splits up, the prediction region become wider by covering almost all possible routes of the specific vessel type. Figures 4 depict abnormal vessel traffic cases. Both methods, LSTM prediction region and wild bootstrapping has classified those as abnormal. Figure 4(a) shows anomalous case where a cargo vessel unexpectedly turned around by changing direction 180 degrees due captains decision to return port for repair a engine malfunction. The first 50 vessel's navigational vectors are given into model's input. The model by using both methods had predicted regions where true position of the vessel is expected. Because vessel made sharp change in direction, the true values were outside of predicted regions by both methods. The same figure 4(a) shows that true 25th and 50th vessel positions (black triangles) are outside prediction regions by LSTM region prediction method (red rectangles) and wild bootstrapping method (blue ellipses). The methods classify such vessel traffic as abnormal because it does not fall in $\alpha = 95\%$ prediction region. Figure 4(b) depicts a drifting cargo ship due of broken engine. It is actual navigational vectors are outside of prediction region. Figures 4(c) and 4(d) show other abnormal cases. The first is unexpected turn to minor port, and second unplanned stop due engine failure.

Table 1 summarizes results of marine traffic prediction and its evaluation. LSTM prediction region method performs better with the parameter $\alpha = 0.95$ for both training and test data set. Wild bootstrapping performs much worse as set α . Also, due to nature of algorithm, this method cannot be trained and tested on only test data set, it needs whole set to calculate predictions.

Table 1. Detected abnormal traffic results.

Method	Sequences		Ratios		MSE
	Normal	Abnormal	Normal	Abnormal	
Training data set					
LSTM prediction region	19922	1049	0.95	0.05	$1.21 \cdot 10^{-3}$
Wild bootstrapping	18035	2936	0.86	0.14	$1.22 \cdot 10^{-3}$
Test data set					
LSTM prediction region	4980	263	0.95	0.05	$1.22 \cdot 10^{-3}$
Wild bootstrapping	4352	891	0.83	0.17	$1.24 \cdot 10^{-3}$

4 Conclusions

This paper investigates vessel movement prediction and prediction evaluation techniques that can be applied for traffic abnormality detection. The literature review revealed that the most algorithms of trajectory prediction are supervised or semisupervised. The authors of the paper proposes new unsupervised trajectory point prediction and prediction regions at the arbitrary probability. The paper depicts two methods – the LSTM prediction region learning and the wild bootstrapping.

The LSTM method is based on learning of prediction region with the view to reach required confidence level by learning the parameters of custom loss function. The prediction region is defined by multivariate LSTM models according to different loss functions for learning the upper and lower bounds that produce prediction region of trajectory point in the shape of hyperrectangle. By investigating experimentally it was observed that the 95 % LSTM prediction region is wider than that obtained by wild bootstrapping technique. And it is assumed that the traffic outside the prediction region is abnormal.

The second proposed method is based on statistical wild bootstrap approach that estimates 95 % prediction region. Nevertheless, during the testing, it was noticed that only 83 % of true vessel trajectory point values are inside wild bootstrapping prediction region. Thus, the method provides narrower confidence regions than those obtained by LSTM. The approach is recommended to use where strict control of marine traffic is required such as sea ports, seaport surroundings, or other sea regions with limitations induced by designated geographical locations.

Results shows that both the LSTM and wild bootstrapping algorithms for estimation of prediction regions can be used for abnormal marine traffic detection. The experiments with the data shows that algorithms, by evaluating prediction region, can detect different types of abnormal marine traffic such as: vessel slowdown, turn around, sharp direction change, unplanned stop, traffic not on seaway, etc. in an unsupervised manner.

Acknowledgment. The authors are 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.

References

1. Danish Maritime Authority, Historical AIS data, <https://www.dma.dk/SikkerhedTilSoes/Sejladsinformation/AIS/Sider/default.aspx>, 2020.
2. V. Chew, Confidence, prediction, and tolerance regions for the multivariate normal distribution, *J. Am. Stat. Assoc.*, **61**(315):605–617, 1966, <https://doi.org/https://doi.org/10.2307/2282774>.
3. C. Chiu, T.N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, N. Jaitly, B. Li, J. Chorowski, M. Bacchiani, State-of-the-art speech recognition with sequence-to-sequence models, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada*, IEEE, 2018, pp. 4774–4778, <https://doi.org/10.1109/ICASSP.2018.8462105>.
4. P.N.R. Chopde, M.K. Nichat, Landmark based shortest path detection by using A* and haversine formula, *International Journal of Innovative Research in Computer and Communication Engineering*, **1**(2):298–302, 2013.
5. 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.
6. N. Cruz, L.G. Marín, D. Sáez, Neural network prediction interval based on joint supervision, in *2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil*, IEEE, 2018, pp. 1–8, <https://doi.org/10.1109/IJCNN.2018.8489264>.
7. N. Cruz, L.G. Marín, D. Sáez, Prediction intervals with LSTM networks trained by joint supervision, *2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary*, **14-19 July 2019**:1–8, 2019, <https://doi.org/10.1109/IJCNN.2019.8852215>.
8. P.K. Diederik, B. Jimmy, Adam: A method for stochastic optimization, in Yoshua Bengio, Yann LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015*, Conference Track Proceedings, 2015, <http://arxiv.org/abs/1412.6980>.
9. P. Fu, H. Wang, K. Liu, X. Hu, H. Zhang, Finding abnormal vessel trajectories using feature learning, *IEEE Access*, **5**:7898–7909, 2017, <https://doi.org/10.1109/ACCESS.2017.2698208>.
10. A. Graves, J. Schmidhuber, Offline handwriting recognition with multidimensional recurrent neural networks, in D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), *Advances in Neural Information Processing Systems (NIPS 2008), Vol. 21*, Curran Associates, 2009, <http://papers.nips.cc/paper/3449-offline-handwriting-recognition-with-multidimensional-recurrent-neural-networks.pdf>.
11. M.J. Hausknecht, P. Stone, Deep recurrent Q-learning for partially observable MDPs, in *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15)*, Arlington, VA, 2015, <http://www.cs.utexas.edu/users/ai-lab/hausknecht:sdmia15>.
12. C.-Y. He, Z. and Chow, J.-D. Zhang, STNN: A spatio-temporal neural network for traffic predictions, *IEEE Trans. Intell. Transp. Syst.*, **1**(1):1–10, 2020, <https://doi.org/10.1109/tits.2020.3006227>.

13. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9**(8):1735–1780, 1997, <https://doi.org/10.1162/neco.1997.9.8.1735>.
14. M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *AIChE J.*, **37**(2):233–243, 1991, <https://doi.org/10.1002/aic.690370209>.
15. D. Li, L. Li, X. Li, Z. Ke, Q. Hu, Smoothed LSTM-AE: A spatio-temporal deep model for multiple time-series missing imputation, *Neurocomputing*, **411**:351–363, 2020, <https://doi.org/10.1016/j.neucom.2020.05.033>.
16. L.G. Marin, N. Cruz, D. Saez, M. Sumner, Nunez A., Prediction interval methodology based on fuzzy numbers and its extension to fuzzy systems and neural networks, *Expert Syst. Appl.*, **119**:128–141, 2019, <https://doi.org/10.1016/j.eswa.2018.10.043>.
17. World Weather Online, World weather online meteorological data of Danish waters region, <https://www.worldweatheronline.com/>, 2020.
18. A. Sagheer, M. Kotb, Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems, *Sci. Rep.*, **9**:19038, 2019, <https://doi.org/10.1038/s41598-019-55320-6>.
19. A. Sidibé, G. Shu, Study of automatic anomalous behaviour detection techniques for maritime vessels, *J. Navig.*, **70**(4):847–858, 2017, <https://doi.org/10.1017/S0373463317000066>.
20. S. Tian, X. Zhang, Y. Zhang, Z. Cao, W. Wei, Spatio-Temporal position prediction model for mobile users based on LSTM, in J. Ma, L.T. Yang (Eds.), *Proceedings of the Eleventh International Conference on Parallel and Distributed Systems*, 20–22 July 2005, Fukuoka, Japan, IEEE, 2019, pp. 967–970, <https://doi.org/10.1109/ICPADS47876.2019.00146>.
21. J. Venskus, M. Kurmis, A. Andziulis, Z. Lukošius, M. Voznak, Bykovas, 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)*, year, Volume IEEE, 2015, <https://doi.org/10.1109/SPECTS.2015.7285281>.
22. J. Venskus, 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*, Belarusian State University, Minsk, 2019, pp. 307–310.
23. J. Venskus, 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, 25–27 September 2019, Granada (Spain)*, Godel Impresiones Digitales S.L., Granada, 2019, p. 38.
24. J. Venskus, P. Treigys, J. Bernatavičienė, J. Markevičiūtė, 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 Univ. Press, Vilnius, 2019, p. 89, <https://doi.org/10.15388/Proceedings.2019.8>.
25. J. Venskus, P. Treigys, J. Bernatavičienė, G. Tamulevičius, V. Medvedev, Real-time maritime traffic anomaly detection based on sensors and history data embedding, *Sensors (Basel)*, **19**(17), 2019, <https://doi.org/10.3390/s19173782>.

26. J. Venskus, P. Treigys, J. Bernatavičienė, V. Medvedev, M. Voznak, M. Kurmis, V. Bulbenkienė, Integration of a self-organizing map and a virtual pheromone for real-time abnormal movement detection in marine traffic, *Informatica (Vilnius)*, **28**(2):359–374, 2017, <https://doi.org/10.15388/Informatica.2017.133>.
27. Z. Wan, J. Chen, A.E. Makhoulfi, D. Sperling, Y. Chen, Four routes to better maritime governance, *Nature*, **540**(7631):27–29, 2016.
28. J. Will, L. Peel, C. Claxton, Fast maritime anomaly detection using Kd-tree Gaussian processes, in *Proceedings of the IMA Maths in Defence Conference, Swindon, UK*, 2011.