Vilniaus universitetas
Duomenų mokslo ir skaitmeninių technologijų institutas
LIETUVA

INFORMATIKOS INŽINERIJA (T007)

DOKTORANTO MOKSLINĖ ATASKAITA UŽ 2018/2019 METUS

Julius Venskus

2019 m. spalis

Mokslinė ataskaita DMSTI-DS-T007-19-14

VU Duomenų mokslo ir skaitmeninių technologijų institutas, Akademijos g. 4,
Vilnius LT-08412
www.mii.lt

Meteorological Data Influence on Missing Vessel Type Detection Using Deep Multi-stacked LSTM Neural Network

J. Venskus, P. Treigys
Vilnius University
Vilnius, Lithuania e-
mail: julius.venskus@mii.vu.lt

1.1 *Abstract*

Highly-loaded seaports have extremely complex and intensive marine vessel traffic, which generates large volumes of traffic data. Meteorological conditions and maritime vessel type influence maritime traffic and they must also be taken into account in order to train the model capable of recognizing the abnormal movement of the sea transport. Real data often misses some data values, such as type of vessel or its status. This paper reviews method of obtaining vessel traffic and meteorological data and filling missing vessel type data in Rotterdam port region. A deep multi-stacked LSTM neural network model is trained to fill the missing vessel type data. This model is trained with available vessel type data and used to predict missing values. This paper describes creation and evaluation of this model. Results of the experiment show it is expedient to use traffic data of a vessel in conjunction with meteorological data.

2 Introduction

Maritime transport is one of the most important and intense sectors of human activity, accounting for about 90% of total trade. The high volume of vessel traffic generates large amounts of data, which overload various information systems and sensors[3]. Assistive systems are developed to facilitate the task, which extract the necessary information from the big data. One of the systems is an unusual traffic detection system, which requires full data for accurate detection. Unfortunately, the data that comes from different systems such as AIS, radars or satellite systems, is not full at all times[1]. The lack of such data prevents the creation of a sufficiently accurate model for detection of unusual vessel traffic. It is therefore necessary to develop smart systems for filling in the missing data, especially with the increased development of new methods for the detection of unusual traffic, which is essential for safety at sea[5]. This article offers a way to fill in the missing data for missing vessel types, which would allow for improved prediction of abnormal maritime traffic. The first part of the article introduces the developed method used to fill in the missing vessel type information in the data, and the second part describes the experiments with this method using vessel traffic data in the Rotterdam harbour. This research is continuous work in field of abnormal maritime traffic detection[4].

3 Proposed Method

Deep neural neural network: The main purpose of the model being developed is to determine the type of vessel by the available or received sets of vessel positions so that the missing information fields can be filled. The model input consists of a sequence of vessel position vectors, and the model predicts the type of vessels sailing under these sets. The model for vessel type prediction uses historical sets of vessel position vectors sorted by time, which can be represented as follows: $X_T = [X_{T-(n-1)},X_{T-(n-2)},,X_{T-1},X_T]$, . Where X_T is the set of the vessel's positions, T is the sequence number of a vessel set, which was received at a certain time, n is the predefined length of the set. The input vector X consists of the positioning elements of the vessel, such as the latitude, longitude, heading, speed, time, state reported by the vessel, weather conditions in the geographical location. We can describe the full input vector as a matrix:

$$X_T^p = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^p \end{bmatrix} = \begin{bmatrix} X^1_{T-(n-1)} & X^1_{T-(n-2)} & \cdots & X^1_{T-1} & X^1_T \\ X^2_{T-(n-1)} & X^2_{T-(n-2)} & & X^2_{T-1} & X^2_T \\ & \vdots & \ddots & \vdots & \\ X^p_{T-(n-1)} & X^P_{T-(n-2)} & \cdots & X^p_{T-1} & X^p_T \end{bmatrix},$$

Where p is the number of elements in the vessel's position vector. The output vector consists of the predicted distribution of probability classes of vessel types calculated by Softmax function. LSTM Deep Neural Network[2] with fully connected multilayer percepron is used to train the model at work. The simplified network architecture is shown in Figure 1. The deep network architecture diagram shows a network structure consisting of 6 constituent layers. The first layer is input layer In with a number of inputs that equals to the length of the vessel's position sequence n. The input layer is connected in series to the first n cells from A1 to an in LSTM (A) layer. The LSTM layer may have more than n cells. The total number of cells is expressed in k when k n. LSTM (A) layer is connected in series to the LSTM (B) layer. Each output of layer A is connected to Layer B inputs. The total number of cells in LTSM (B) is expressed in k. Both LSTM layers use ReLu activation function. The last cell in B is connected to the multilayer fully connected layer of perceptron. The layer of perceptron consists of two hidden layers of neurons and one output layer. The hiddens layers use ReLu activation function. A number j of outputs constitutes an output layer where each output describes the probability of a particular class classification, which is calculated by Softmax function. Adam's stochastic optimizer with a training factor $\alpha = 0.001$ and a decay factor $\delta = 10^{-6}$ are used for network training. The termination of epoch training cycles is set in accordance with the validation set. The training uses the Sparse Categorical Cross Entropy[2] for loss function. **Data preparation:** Duplicate position entries are cleared out and then data is parsed based on desired data types. The same actions are performed to meteorological data. Technical data fields of a vessel are assigned to each position vector of a vessel based on vessel MMSI identificator. Meteorological data is assigned to a position data vector by using the method of the closest neighbor, depending on the closest time and geolocation of the forecast. Model train-

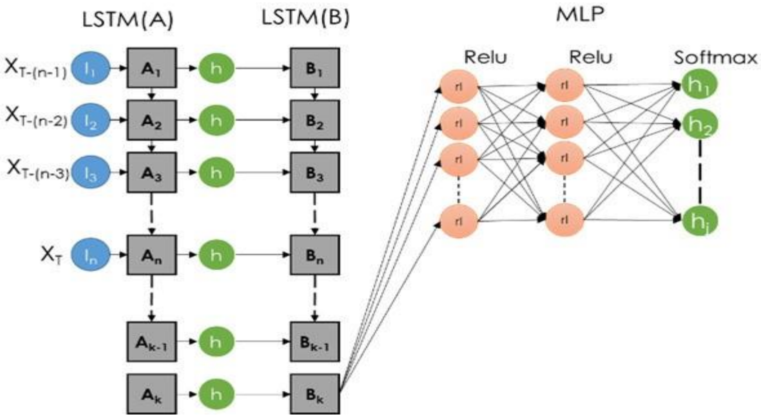

Figure 1: Multi-Stacked LSTM deep neural network architecture

ing data is formed by joining all the data to one vector. **Creation of Vector Sets:** After the preparation of data the vector sets that will be provided to train, validate and test the model, are further formed. The data sets must correspond to the model input matrix, which is described by formula above. To do this, the available vessel data is grouped by their MMSI identifier. All consecutive position of a vessel is cut in sequences of 12 positions by step of 3 positions. All these formed sequences are used to construct training matrix described in this article above.

4 Experiment and Results

To test proposed method Rotterdam harbour area was chosen. The data for model training are collected from several sources such as AIS vessel traffic monitoring system, vessel parameter information system, meteorological observation system, and geographic information system. This information comes from several data sources. The marine traffic data was colected from shipfinder.co: Colected data is: geolocation, speed, direction and type of vessel, length, width, draft, etc. The Meteorological data colected from worldweatheronline.com, provides meteorological data in given geolocations: wind direction and speed, wave, swell and other data of a particular location in 3h intervals. Two separate set was formed to test influence of meteorogical data. One set with meteo data, another without. A total of $2.90 * 10^7$ vessel traffic vectors were collected in one set from the Rotterdam harbour from November 1, 2018 till November 30, 2018, of which $2.78*10^7$ do not have information about vessel type. This represents 95.88% of all available data. A set with vessel type information consists of $1.195 * 10^6$ vessel traffic vectors from Rotterdam harbour. These vectors were collected and created using the methods mentioned above, and they constitute 4.12% of all data. The data are randomly divided into three sets: 50% of the data are used for training, 30% for validation, and 20% for testing. Training data set is used to train models. Validation Set - is designed to select the number of LSTM layers in the model and LSTM cells in the layer. The test set is used to evaluate accuracy of the final model. In this

Table 1: Trend of classification accuracy for different network settings

	Meteorological data excluded		Meteorological data included	
Layers	Cells	Accuracy	Cells	Accuracy
2	245	0.78	290	0.78
3	215	0.79	265	0.81
4	195	0.77	250	0.93

article precision, recall, and accuracy are calculated using a test set in order to evaluate the accuracy of the classifier for different numbers of deep multi-stacked LSTM neural network layers and cells. Table 1 first part shows the results of the experiment for different values of the model parameters without meteorological data. We see that the best result was achieved using 3 LSTM layers made of 215 cells. Table 1 second part shows the results of the experiment with meteorological data. Best result was achieved using 4 LSTM layers with 250 cells.

5 Conclusion

According to the results of the experiment, the proposed method of combining vessel traffic data with meteorological data leads to an improved classification. Based on the results the best model configuration is chosen, then checked using continued data with classification accuracy 0.93, recall 0.92, and precision 0.93.

References

[1] Alvarez A.A. et al. (2016). Mining Vessel Tracking Data for Maritime Domain Applications. *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference.* pp. 361–367.

[2] Ergen T., Kozat S.S. (2018). Online Training of LSTM Networks in Distributed Systems for Variable Length Data Sequences. *IEEE Transactions on Neural Networks and Learning Systems.* Vol. **29**. pp. 5159 - 5165.

[3] Sun F., Deng Y., and Deng F. (2015). Unsupervised maritime traffic pattern extraction from spatio-temporal data. 11th International Conference on Natural Computation (ICNC). pp. 1218-1223.

[4] Venskus J. at al. (2017) Integration of a self-organizing map and a virtual pheromone for real-time abnormal movement detection in marine traffic. *Informatica. Vilniaus Universitetas.* Vol. **28**. pp. 359-374.

[5] Zhen R., et al. (2017). Maritime Anomaly Detection within Coastal Waters Based on Vessel Trajectory Clustering and Na¨ıve Bayes Classifier. *The Journal of Navigation.*, pp. 1–23.

sensors

MDPI

6 *Article*

Real-Time Maritime Traffic Anomaly Detection Based on Sensors and History Data Embedding

7 **Julius Venskus[†,‡], Povilas Treigys[*,†,‡], Jolita Bernatavicˇiene˙[†,‡], Gintautas Tamulevicˇius[†,‡] and Viktor Medvedev[†,‡]**

Institute of Data Science and Digital Technologies, Vilnius University, Akademijos str. 4, LT-08412 Vilnius, Lithuania

***** Correspondence: povilas.treigys@mif.vu.lt

† Current address: Akademijos str. 4, LT-08412 Vilnius, Lithuania ‡ These authors contributed equally to this work.

Received: 12 July 2019; Accepted: 28 August 2019; Published: date

Abstract: The automated identification system of vessel movements receives a huge amount of multivariate, heterogeneous sensor data, which should be analyzed to make a proper and timely decision on vessel movements. The large number of vessels makes it difficult and time-consuming to detect abnormalities, thus rapid response algorithms should be developed for a decision support system to identify abnormal movements of vessels in areas of heavy traffic. This paper extends the previous study on a self-organizing map application for processing of sensor stream data received by the maritime automated identification system. The more data about the vessel's movement is registered and submitted to the algorithm, the higher the accuracy of the algorithm should be. However, the task cannot be guaranteed without using an effective retraining strategy with respect to precision and data processing time. In addition, retraining ensures the integration of the latest vessel movement data, which reflects the actual conditions and context. With a view to maintaining the quality of the results of the algorithm, data batching strategies for the neural network retraining to detect anomalies in streaming maritime traffic data were investigated. The effectiveness of strategies in terms of modeling precision and the data processing time were estimated on real sensor data. The obtained results show that the neural network retraining time can be shortened by half while the sensitivity and precision only change slightly.

Keywords: streaming sensors data; neural network retrain time; model sensitivity and precision; marine traffic anomaly detection; SOM data batching

7.1 *1. Introduction*

The maritime industry is an important part of the global trade system with a growing volume, intensity, and needs. In 2018, 1.9 billion tons of goods were transported as part of EU short sea shipping [1]. This is 3.2% more in comparison with 2016. Totally, more than 90% of cargo is carried by sea transport [2].

Such growth presents some challenges in the industry. Increasing intensity of maritime traffic raises the need for incident prevention-oriented traffic control. The maritime anomaly or abnormal movement detection is one of the control techniques. It is based on vessel trajectory analysis and search of irregular, illegal, and other anomalous appearances in trajectory data [3]. A maritime trajectory can include vessel identification data, traffic parameters (e.g. speed and rotation), auxiliary data (e.g., meteorological data) for a vessel, and such dataset presents a large-scale, complex data structure. Automated data gathering systems (e.g., Automatic Identification System) return larger and larger trajectory datasets, which are challenging for human-based analysis and anomaly detection [4]. Nowadays, machine learning-based data analysis and mining techniques is a natural choice for this type of task: the obtained structure of data, the extracted information, detected data regularities could help to estimate vessel movement and make some safety decision, to enable the automatic anomaly detection even. For real-world applications, a challenge of real-time operation, data generalization arises. Movement anomalies are detected as history-based deviations of vessel’s trajectory data, which can be problematic considering massive trajectory data streams. In this case, constant estimation of historical and context data means permanent need for system retraining. Full retraining is a time- and power-consuming process; therefore, some techniques of additional or adaptive training would be preferred: rapid self-learning algorithms have to be developed to detect the abnormal movement in stream data.

The paper is organized as follows. Section 2 presents the problem of abnormal movement detection in maritime traffic data and gives the state-of-the-art problem solutions. In Section 3, the motivation of this paper is presented: two retraining strategies are introduced for neural network-based real-time maritime anomaly detection. The results of experimental research of these strategies are given in Section 4. The investigation results are concluded in Section 5.

7.2 2. Review

In this section, we present maritime anomaly detection task and review some recent research results in this area.

The abnormal vessel movement can be defined as an unreasoned movement deviation from the sea lanes, trajectory, speed or other traffic parameters [5]. As most vessels have the Automated Identification System (AIS) installed, giving the static and dynamic information about the vessel movement, the detection of traffic anomaly comes as the task of data analysis and outlier detection. In addition, different sensor systems can be connected to the AIS. Traffic data are analyzed in point-based or trajectory-based manner [6].

In the first case, every single data point (message from the vessel to the AIS) or a group of them is treated as an independent point. For this purpose, the analyzed geographical area is subdivided into independent cells with related AIS messages. These data points in the grid are analyzed using so-called signature-based or rule-based techniques. The idea of these techniques is the employment of various association rules to detect specific movement changes [7]. Zhu applied database management, data warehouse, and data mining technologies to analyze AIS data [8]. Deng [9] extended the features and inserted time stamps. These extensions enable employing Markov model for supplementation of rules. While declaring the point-based analysis, Pallotta et al. [10] proposed to use a sliding time window to estimate the relationship between successive AIS data points. The obtained waypoints are clustered using Density-Based Spatial Clustering of Applications with Noise methodology and employed for anomaly detection and movement prediction. Despite the claims about point-based analysis, the authors implemented the idea of updating the traffic knowledge from the input of AIS messages and the use of historical knowledge. The same clustering methodology was explored in [11].

Here, the historical spatiotemporal data are analyzed to detect waypoints of routes.

The main weakness of point-based techniques is the analysis of movement short-term history or disregard of history even. The planned and purposing vessel movement should generate highly-correlated AIS data, and this can be used for movement anomaly detection. On the other hand, a limited number of analyzed data points means 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.

Trajectory-based techniques treat the entire traffic data sequence as a whole. Several research directions are analyzed in the literature related to the analysis of vessel trajectories: maritime traffic pattern mining, ship collision risk assessment [12], maritime anomaly detection [13–15], identification of the types of ships [16], and combating abalone poaching [17].

In the case of trajectory-based detection, models of normal movement are created (using the entire trajectory data, not part of it) and the anomalies are detected as movement data inadequacy to the model. Thus, these techniques are characterized by having a huge amount of AIS data to analyze.

This property requires some data pre-processing such as compression or clustering.

In [18], a piece-wise linear segmentation is applied to compress the data of vessel trajectories, and then the similarity of trajectories (for detection of anomalies) is performed using alignment kernels (dynamic time warping and edit distances, namely). The model by Lei [13] defines spatial, sequential, and behavioral features of the vessel movement. The movement anomaly is detected as the outlying features of the trajectory model, and the degree of suspiciousness is evaluated. The geometrical properties of the trajectory are employed in [19]. Here, the vessel trajectory is compared with the graph search-based path and the difference is estimated by a final score. The threshold value of the score is employed as the decision and labeling value. Another trajectory-based analysis techniques can be found in [20–22].

Analysis of the entire trajectory gives the advantage of the historical movement data, which can be essential for anomaly detection. However, full data analysis requires much more complicated algorithms such as neural networks. This complicates the application of trajectory-based analysis for real-time tasks. In addition, such algorithms are sensitive to missing data (e.g., lost AIS messages).

A comprehensive and categorizing review on maritime anomaly detection can be found in [5,15,23].

Analysis of full trajectory data and anomaly detection would require data-driven approaches such as artificial neural network-based or statistical methods. These approaches can perform in an unsupervised or semi-supervised manner (i.e., they do not need labeled data) and can cope with large amounts of data. The issue of real-time calculations should be solved using the idea of incremental modeling (retraining, re-estimating, etc.): the model of vessel movement should be updated concerning recent data to avoid of complete remodeling or model retraining.

7.3 3. Motivation

The vessel movement (normal or abnormal) can be treated differently regarding the sea region where the movement is observed. For example, if the ship is quite distant from the seaport, then even high decline from its course cannot be indicated as an anomaly: weather condition, stormy sea, etc. may have a great influence on vessel trajectory. On the other hand, if vessel movement is observed at the seaport surroundings, even a small deviation from the course may be thought as abnormal vessel activity. To this purpose, the method used for traffic anomaly detection has to have a feature that allows different region scaling at different maritime traffic observation areas. The self-organizing map (SOM) method has such a scaling property. SOM is a neural network-based method that is trained in an unsupervised way using a competitive learning [24–27]. The neural network can be used for both visualization and clustering of multidimensional data [28].

In the previous research [29], the modified SOM algorithm for maritime vessel movement data classification into normal and abnormal classes is presented. The modification is achieved by incorporating virtual pheromone intensity calculations at the last epoch of model training. During the model validation stage, the pheromone intensity threshold is established by applying a gradient descent method. The dependence of the network neighboring function on the classification results was investigated; the best classification accuracy qA achieved using the Mexican hat neighboring function. The influence of different SOM grid dimensions on the classification results of the proposed algorithm has been investigated. It was proved experimentally that the algorithm achieved the best precision using grid dimension 25 × 25. This knowledge was used as a starting point for the network data batching and training strategies investigation presented in this paper.

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 retrain quite straightforward: the more vessel movement data that 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 [30]. Other authors proposed neural networks retraining strategies to build compact neural network models with less memory usage and faster inference speed [31]. Recently, the SOM neural network is being used to build datasets used in deep neural network model retraining [30,32] or is used as a part of deep neural network model [33]. Different areas of applications of the SOM algorithm depicts the necessity to investigate more thoroughly algorithm effectiveness 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 has to be explored [34]. In this article, the authors introduce two neural network retrain strategies and compare the results with the standard procedure of neural network model experimental investigation (so-called Strategy I).

- 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. In this paper, it is used as a reference with the view to compare retrain Strategies II and III introduced by the authors.

- Strategy II presents algorithm performance while using pre-trained model parameters on previously trained data with the newly arriving data batches.
- Strategy III presents different data batch shuffling techniques and the use of previously pre-trained model parameters.

All three strategies investigate 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 manned based on the up-to-date data it receives.

7.4 4. Experiments

In this section, we present a detailed description of the SOM network retraining strategies and results of the experiments using real datasets.

7.4.1 4.1. Data Preparation

The detailed description of the previous study of SOM size and modification by introducing the SOM evaporation functions are presented in [29]. Data from the region of medium maritime traffic at the Klaipeda seaport were selected for the analysis of the proposed retraining strategies of the SOM network. During the experiments, two datasets were used: Cargo vessels and Passenger vessels. Each item (point) of a vessel's streamed data is described by longitude, latitude, heading, vessel speed, wind direction, wind speed, wave direction, and wave height values. The Cargo dataset is represented by 180,300 and the Passenger dataset is described by 43,879 vessel movement observation items that were registered in a streamed manner. All experiments in this section were carried out with the Cargo dataset; afterwards, the data batching strategies were tested on the Passenger dataset.

First, 20% of the Cargo vessel dataset was randomly selected for the general model error evaluation. Then, the resulting 80% of the dataset items were used for the data batching strategy investigation. These 80% of data items were split into 20% for strategy testing, and 80% for T1, T2, and T3 data batch splitting (see Figure 1) 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 network retraining strategies and learning rate parameter selection. The scheme of data split is shown in Figure 1.

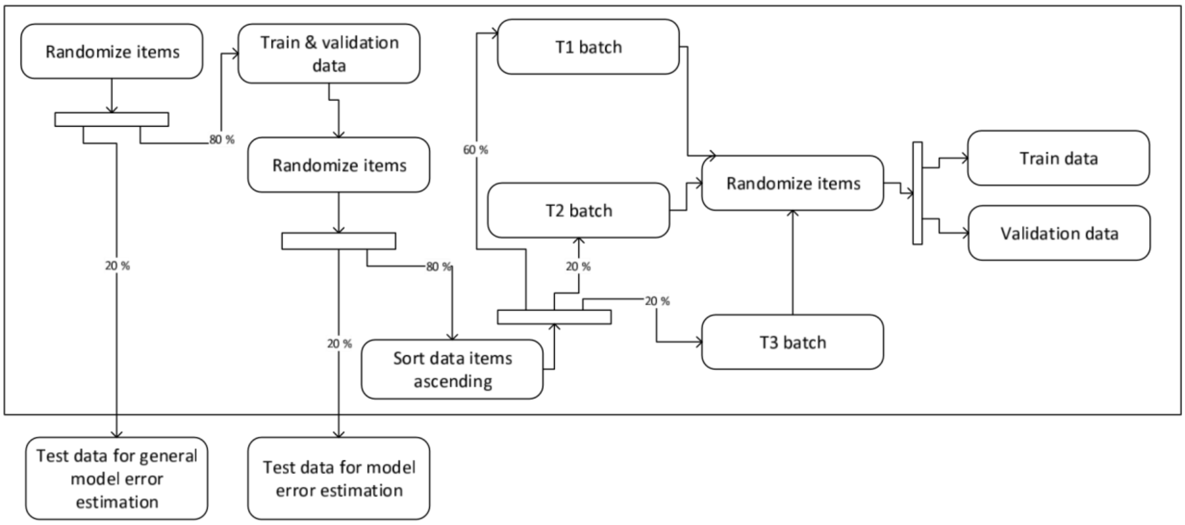

Figure 1. Data split scheme.

All data items were sorted in ascending order with respect to data sending time. The SOM network of size 25 × 25 was taken according to the SOM size investigation published in [29].

7.4.2 4.2. Training Strategies of the SOM Network

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 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. 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 1). 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 dataset, which allowed evaluating the general model error. The best-obtained model characteristics with model test dataset are presented in Table 1 (bold line).

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 2. The time needed for the algorithm retraining was 40,769 s. Strategy II. The initial algorithm was 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. The performance of the investigated network on repetitive Strategy II (using only T1 dataset) model evaluation and testing is presented in Table 3. The line marked in bold shows the best network obtained. Quite small deviations of the precision and the sensitivity rates show the network stability. 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 the additional experiment show that the best performance network was obtained with learning rate 0.025.

The statistics (model test error and general model error evaluation) of the best model data are presented in Table 4. The time needed for model training was 18,229 s.

Table 1. Selection of learning rate.

Learning Rate	TP	FP	TN	FN	Precision	Sensitivity
0.005	924	519	26,648	757	0.6403	0.5497
0.010	943	505	26,662	738	0.6512	0.5610
0.015	957	498	26,669	724	0.6577	0.5693
0.020	963	487	26,680	718	0.6641	0.5729
0.025	968	478	26,689	713	0.6694	0.5758
0.030	976	471	26,696	705	0.6745	0.5806
0.035	986	468	26,699	695	0.6781	0.5866
0.040	998	461	26,706	683	0.6840	0.5937
0.050	1025	445	26,722	656	0.6973	0.6098
0.060	1066	413	26,754	615	0.7208	0.6341
0.070	1109	394	26,773	572	0.7379	0.6597
0.100	1197	303	26,864	484	0.7980	0.7121
0.200	1431	135	27,032	250	0.9138	0.8513
0.300	1486	81	27,086	195	0.9483	0.8840
0.400	1500	55	27,112	181	0.9646	0.8923
0.500	**1510**	**52**	**27,115**	**171**	**0.9667**	**0.8983**
0.600	1507	54	27,113	174	0.9654	0.8965
0.700	1502	59	27,108	179	0.9622	0.8935

Table 2. Training Strategy I performance at learning rate 0.5.

Stage	TP	FP	TN	FN	Precision	Sensitivity
Testing (model error)	1510	52	27,115	171	0.9667	0.8983
Testing (general error)	1868	69	33,890	233	0.9644	0.8891

Table 3. Strategy II performance on model test data.

No.	TP	FP	TN	FN	Precision	Sensitivity
1	1364	241	26,926	317	0.8498	0.8114
2	1329	280	26,887	352	0.8260	0.7906
3	1359	252	26,915	322	0.8436	0.8084
4	1364	274	26,893	317	0.8327	0.8114
5	1356	253	26,914	325	0.8428	0.8067
6	1335	253	26,914	346	0.8407	0.7942
7	1314	251	26,916	367	0.8396	0.7817
8	1332	258	26,909	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 4. Retraining Strategy II performance at learning rate 0.025.

Stage	TP	FP	TN	FN	Precision	Sensitivity

Testing (model error)	1500	98	27,069	181	0.9387	0.8923
Testing (general error)	1836	122	33,837	265	0.9377	0.8739

Strategy III. The scheme of the model training validation and testing was similar to that described in Strategy II, except for the following two things. Firstly, from T2 and T3 batches, there were produced four data batches (Tm2–Tm5), each containing one quarter of both T2 and T3 data (see Table 5). 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 the model training data aggregation. For every retraining. 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 were compounded of items from T2 and T3, as shown in Table 5 (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 ships to change their sea routes. Experiments depicted that the best model was obtained with the learning rate being 0.03.

Table 5. Partitioning of dataset (Strategy III).

Data Batches	% of Train and Validation Data	New Data Items	All Data Items
T1	60%	69,235	69,235
Tm2	10%	11,539	23,078
Tm3	10%	11,539	23,078
Tm4	10%	11,539	23,078
Tm5	10%	11,539	23,078

The statistics of the Strategy III best model were obtained using test data for general model error estimation, and the results are presented in Table 6.

Table 6. Retraining Strategy III performance at learning rate 0.003.

Stage	TP	FP	TN	FN	Precision	Sensitivity
Testing (model error)	1527	73	27,094	154	0.9544	0.9084
Testing (general error)	1866	91	33,868	235	0.9535	0.8881

The time needed for the algorithm retraining was 27854 s. The summary of relative time required for the training Strategies I–III is presented in Table 7.

Table 7. Retraining Strategies I–III performance on Cargo dataset.

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

The same data batching Strategies I–III described above were tested on the Passenger dataset as well. The results are presented in Table 8.

Table 8. Retraining Strategies I–III performance on Passenger dataset.

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

From the results shown in Tables 7 and 8, it can be seen that, by applying different SOM model retraining Strategies, while keeping the same data batch sizes, it is possible to substantially decrease the time for maritime traffic abnormal movement detection while retraining the model 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. The results presented in Table 8 prove the correctness of the training strategies investigation.

7.5 *5. Conclusions*

This paper extends the previous study on a self-organizing map application, which is trained in an unsupervised way using competitive learning, for processing of sensors stream data in order to detect abnormal vessel movement in maritime traffic. Different strategies for the unsupervised retraining of the SOM network to classify maritime vessel movement data into normal and abnormal classes were presented and investigated. The data batching strategies ensure high precision of the algorithm by introducing a huge amount of new data on vessel movements. Two different unsupervised SOM network retraining strategies for maritime vessel movement data classification into normal and abnormal classes were proposed and investigated. The experimental research depicted promising results. The study showed that the SOM network can be retrained in half the time by only applying different train/validation and test datasets. The initial results depict that the obtained speed-up in data processing time maintains precision and sensitivity, varying not more than 3% in unusual maritime traffic detection.

The results of the experiments show that:

- If the model is trained from 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 0.889 at learning rate 0.5.
- If the model is trained on top of the pre-trained model weights, the precision and sensitivity slightly drop, but the training time decreases by half at learning rate 0.025.
- If the model is trained on top of the pre-trained model weights and the newly arrived data batch is proportionally mixed with those winning neurons, training time can be decreased by one third

  while keeping almost the same results as depicted previously at learning rate 0.03.

The independent experiment on unseen dataset confirmed the results correctness and allowed concluding that, by applying batched data approach for SOM retraining on the pre-trained model, network training can be shortened to half the time by selecting learning rate parameter from the interval [0.025;0.03] while maintaining the model sensitivity and precision with only minor changes.

Author Contributions: Conceptualization: P.T. and J.V.; Methodology: J.V. and J.B.; Software: J.V.; Validation: V.M. and G.T.; Investigation: J.V. and V.M.; Data curation: J.V.; Writing—original draft preparation: J.V.; Writing—review and editing: G.T., J.B.; and Supervision: P.T.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

8 References

1. The European Commission. Maritime Transport Statistics-Short Sea Shipping of Goods.2019. Available online: https://ec.europa.eu/eurostat/statistics-explained/index.php/Maritime_transport_ statistics_-_short_sea_shipping_of_goods (accessed on 29 August 2019).

2. Wan, Z.; Chen, J.; Makhloufi, A.E.; Sperling, D.; Chen, Y. Four routes to better maritime governance. *Nature* **2016**, *540*, 27–29.
3. Fu, P.; Wang, H.; Liu, K.; Hu, X.; Zhang, H. Finding Abnormal Vessel Trajectories Using Feature Learning. *IEEE Access* **2017**, *5*, 7898–7909.
4. Will, J.; Peel, L.; Claxton, C. Fast maritime anomaly detection using kd-tree gaussian processes. In Proceedings of the IMA Maths in Defence Conference, Swindon, UK, 20 October 2011.
5. Lane, R.O.; Nevell, D.A.; Hayward, S.D.; Beaney, T.W. Maritime anomaly detection and threat assessment. In Proceedings of the FUSION 2010 : 13th International Conference on Information Fusion, Edinburgh, UK, 26–29 July 2010.
6. Sidibé, A.; Shu, G. Study of automatic anomalous behaviour detection techniques for maritime vessels. *J. Navig.* **2017**, *70*, 847–858.
7. Ristic, B. Detecting Anomalies from a Multitarget Tracking Output. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 798–803. doi:10.1109/TAES.2013.130377.
8. Zhu, F. Mining ship spatial trajectory patterns from AIS database for maritime surveillance. In Proceedings of the 2nd IEEE International Conference on Emergency Management and Management Sciences, Beijing, China, 8 August 2011; pp. 772–775. doi:10.1109/ICEMMS.2011.6015796.
9. Deng, F.; Guo, S.; Deng, Y.; Chu, H.; Zhu, Q.; Sun, F. Vessel track information mining using AIS data. In Proceedings of the International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), Beijing, China, 28–29 September 2014; pp. 1–6. doi:10.1109/MFI.2014.6997641.
10. Pallotta, G.; Vespe, M.; Bryan, K. Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. *Entropy* **2013**, *15*, 2218–2245.
11. Arguedas, V.F.; Mazzarella, F.; Vespe, M. Spatio-temporal data mining for maritime situational awareness. In Proceedings of the OCEANS 2015-Genova, Genoa, Italy , 18–21 May 2015; pp. 1–8. doi:10.1109/OCEANS-Genova.2015.7271544.
12. Silveira, P.; Teixeira, A.; Soares, C.G. Use of AIS data to characterise marine traffic patterns and ship collision risk off the coast of Portugal. *J. Navig.* **2013**, *66*, 879–898.
13. Lei, P.R. A framework for anomaly detection in maritime trajectory behavior. *Knowl. Inf. Syst.* **2016**, *47*, 189–214.
14. Zhen, R.; Jin, Y.; Hu, Q.; Shao, Z.; Nikitakos, N. Maritime anomaly detection within coastal waters based on vessel trajectory clustering and Naïve Bayes Classifier. *J. Navig.* **2017**, *70*, 648–670.
15. Riveiro, M.; Pallotta, G.; Vespe, M. Maritime anomaly detection: A review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *2018*, 1–19.
16. Sheng, K.; Liu, Z.; Zhou, D.; He, A.; Feng, C. Research on Ship Classification Based on Trajectory Features. *J. Navig.* **2018**, *71*, 100–116.
17. Dabrowski, J.J.; de Villiers, J.P.; Beyers, C. Context-based behaviour modelling and classification of marine vessels in an abalone poaching situation. *Eng. Appl. Artif. Intell.* **2017**, *64*, 95–111.
18. de Vries, G.K.D.; van Someren, M. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Expert Syst. Appl.* **2012**, *39*, 13426–13439. doi:10.1016/j.eswa.2012.05.060.
19. Soleimani, B.H.; Souza, E.N.D.; Hilliard, C.; Matwin, S. Anomaly detection in maritime data based on geometrical analysis of trajectories. In Proceedings of the 18th International Conference on Information Fusion (Fusion), Washington, DC, USA, 6–9 July 2015; pp. 1100–1105.
20. Radon, A.N.; Wang, K.; Glässer, U.; Wehn, H.; Westwell-Roper, A. Contextual verification for false alarm reduction in maritime anomaly detection. In Proceedings of the IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 1123–1133. doi:10.1109/BigData.2015.7363866.
21. Laxhammar, R.; Falkman, G. Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Ann. Math. Artif. Intell.* **2015**, *74*, 67–94. doi:10.1007/s10472-013-9381-7.
22. Smith, J.; Nouretdinov, I.; Craddock, R.; Offer, C.; Gammerman, A. Conformal Anomaly Detection of Trajectories with a Multi-class Hierarchy. In *Statistical Learning and Data Sciences*; Gammerman, A., Vovk, V., Papadopoulos, H., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 281–290.
23. Cazzanti, L.; Pallotta, G. Mining maritime vessel traffic: Promises, challenges, techniques. In Proceedings of the OCEANS 2015-Genova, Genoa, Italy , 18–21 May 2015; pp. 1–6. doi:10.1109/OCEANS-Genova.2015.7271555.

24. Kohonen, T. The self-organizing map. *Neurocomputing* **1998**, *21*, 1–6.
25. Kayacik, H.G.; Zincir-Heywood, A.N.; Heywood, M.I. A hierarchical SOM-based intrusion detection system. *Eng. Appl. Artif. Intell.* **2007**, *20*, 439–451.
26. Kurasova, O.; Molyte, A. Quality of quantization and visualization of vectors obtained by neural gas and˙ self-organizing map. *Informatica* **2011**, *22*, 115–134.
27. Dzemyda, G.; Kurasova, O.; Žilinskas, J. *Multidimensional Data Visualization: Methods and Applications*; Springer Science & Business Media: Philadelphia, PA, USA, 2012; Volume 75.
28. Medvedev, V.; Kurasova, O.; Bernatavicˇiene, J.; Treigys, P.; Marcinkevicˇius, V.; Dzemyda, G.˙ A new web-based solution for modelling data mining processes. *Simul. Model. Pract. Theory* **2017**, *76*, 34–46.
29. Venskus, J.; Treigys, P.; Bernatavicˇiene, J.; Medvedev, V.; Voznak, M.; Kurmis, M.; Bulbenkien˙ e, V. Integration˙ of a Self-Organizing Map and a Virtual Pheromone for Real-Time Abnormal Movement Detection in Marine Traffic. *Informatica* **2017**, *28*, 359–374.
30. Jafari, A.H.; Hagan, M.T. Application of new training methods for neural model reference control. *Eng. Appl. Artif. Intell.* **2018**, *74*, 312–321.
31. Wang, Z.; Lin, J.; Wang, Z. Accelerating recurrent neural networks: A memory-efficient approach. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 2763–2775.
32. Cannas, B.; Fanni, A.; Pautasso, G.; Sias, G.; others. Disruption prediction with adaptive neural networks for ASDEX Upgrade. *Fusion Eng. Des.* **2011**, *86*, 1039–1044.
33. He, M.; He, D. Deep learning based approach for bearing fault diagnosis. *IEEE Trans. Ind. Appl.* **2017**, *53*, 3057–3065.
34. Bernataviciene, J.; Dzemyda, G.; Bazilevicius, G.; Medvedev, V.; Marcinkevicius, V.; Treigys, P. Method for visual detection of similarities in medical streaming data. *Int. J. Comput. Commun. Control.* **2015**, *10*, 8–21.