



Collision Risk Assessment and Forecasting on Maritime Data (Industrial Paper)

Andreas Tritsarolis
Dept. of Informatics
University of Piraeus
Piraeus, Greece
andrewt@unipi.gr

Nikos Pelekis
Dept. of Statistics & Insurance Science
University of Piraeus
Piraeus, Greece
npelekis@unipi.gr

Brian Murray
Dept. of Energy & Transport
SINTEF Ocean
Trondheim, Norway
brian.murray@sintef.no

Yannis Theodoridis
Dept. of Informatics
University of Piraeus
Piraeus, Greece
ytheod@unipi.gr

ABSTRACT

The wide spread of the Automatic Identification System (AIS) and related tools has motivated several maritime analytics operations. One of the most critical operations for the purpose of maritime safety is the so-called Vessel Collision Risk Assessment and Forecasting (VCRA/F), with the difference between the two lying in the time horizon when the collision risk is calculated: either at current time by assessing the current collision risk (i.e., VCRA) or in the (near) future by forecasting the anticipated locations and corresponding collision risk (i.e., VCRF). Accurate VCRA/F is a difficult task, since maritime traffic can become quite volatile due to various factors, including weather conditions, vessel manoeuvres, etc. Addressing this problem by using complex models introduces a trade-off between accuracy (in terms of quality of assessment / forecasting) and responsiveness. In this paper, we propose a deep learning-based framework that discovers encountering vessels and assesses/predicts their corresponding collision risk probability, in the latter case via state-of-the-art vessel route forecasting methods. Our experimental study on a real-world AIS dataset demonstrates that the proposed framework balances the aforementioned trade-off while presenting up to 70% improvement in R^2 score, with an overall accuracy of around 96% for VCRA and 77% for VCRF.

CCS CONCEPTS

• Information systems → Data mining; Spatial-temporal systems.

KEYWORDS

Maritime Safety, Collision Risk Assessment, Collision Risk Forecasting, Deep Learning

ACM Reference Format:

Andreas Tritsarolis, Brian Murray, Nikos Pelekis, and Yannis Theodoridis. 2023. Collision Risk Assessment and Forecasting on Maritime Data (Industrial Paper). In *The 31st ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '23)*, November 13–16, 2023, Hamburg, Germany. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3589132.3625573>

1 INTRODUCTION

The growing number of ship sensor technologies, such as the Automatic Identification System (AIS), provides a wealth of vessel positioning data well-suited to be used in a wide range of maritime analytics applications [2, 36, 38]. In particular, deep learning, real-time data analytics, and big data processing techniques have attracted the interest of researchers and practitioners in the maritime field who develop models by using large amounts of AIS data to solve various maritime-related problems [2].

Effectively assessing the collision risk in a fleet of monitored vessels is one of the most important aspects of maritime safety [25]. For example, the expected widespread use of unmanned surface vessels (USV) that is foreseen in the near future [4] in order to reduce other modes of transportation, such as road and rail, raises a plethora of maritime safety concerns. In a nutshell, given the current movement status of two vessels, Vessel Collision Risk Assessment (VCRA) and Forecasting (VCRF) aim to assess the current and future, respectively, collision risk of the vessels when they are or are expected to be, respectively, in an encountering process, in terms of their Collision Risk Index (CRI)¹.

Figure 1 illustrates an example, where we depict the vessels' actual (solid black line) and future trajectories (dashed black line), and their corresponding encounterings at t_{now} (hence, VCRA) and up to $t_{now} + \Delta t$ (hence, VCRF). Towards this direction, there is a wide spectrum of state-of-the-art approaches, from mathematical formulae and Fuzzy Logic [31, 41] to Machine Learning / Deep Learning (ML / DL) [18, 21]. In this paper, we propose a framework for addressing VCRA/F based on deep learning, and investigate its application towards ensuring maritime traffic safety.

¹While the problem sounds similar to that of collision avoidance, in terms of decision taking on - high-risk - encountering processes in the foreseeable future, the latter is out the scope of this paper, since it includes the additional stage of recalculating the vessels' itinerary, in order to avoid collision.



This work is licensed under a Creative Commons Attribution International 4.0 License.

SIGSPATIAL '23, November 13–16, 2023, Hamburg, Germany

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0168-9/23/11.

<https://doi.org/10.1145/3589132.3625573>

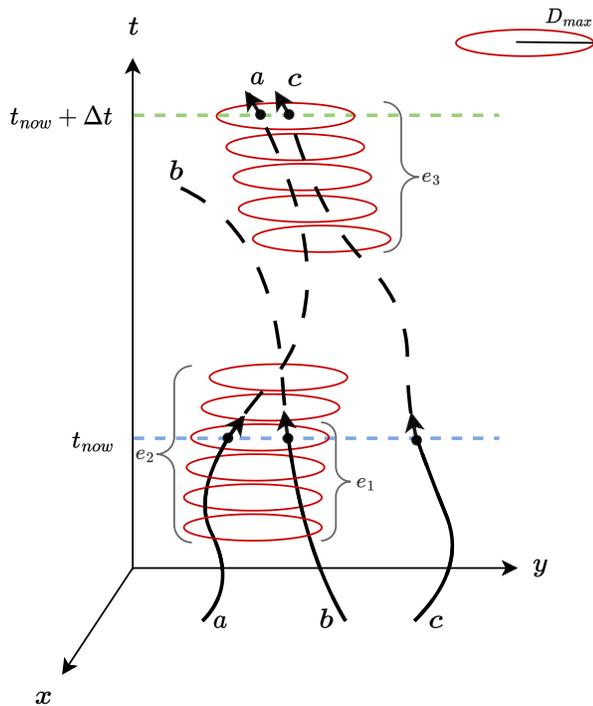


Figure 1: VCRA/F example: VCRA detects encountering e_1 (a, b), whereas VCRF detects encounterings e_2 (a, b) and e_3 (a, c).

In summary, the main contributions of this work are as follows:

- we provide an ML-based method to address the VCRA problem, which outperforms related work;
- we combine our VCRA solution with Vessel Route Forecasting (VRF) in order to address the VCRF problem; and
- we validate our proposed VCRA/F framework over a large-scale real-world AIS dataset and discuss about the explainability of the experimental results.

The rest of this paper is organized as follows: Section 2 discusses related work; Section 3 formulates the problem at hand, whereas Section 4 presents the proposed methodology for VCRA (Section 4.1) and VCRF (Section 4.2); Section 5 presents the setup and the results of our experimental study, while Section 6 discusses about the explainability of the results and the impact of the involved training features in the calculation of CRI; Section 7 concludes the paper, also giving hints for future work.

2 RELATED WORK

2.1 Vessel Collision Risk Assessment

As already mentioned, the current state-of-the-art in VCRA includes formulaic as well as ML-based approaches. The difference between the two aforementioned families of methods is that the former uses kinematic equations combined with ML models (e.g., SVM) while the latter directly leverages ML methods (e.g., CNN) for assessing

the collision risk between two vessels, hereafter called “own” and “target” vessels, respectively².

Focusing on formulaic-based approaches, in [31] the collision risk detection problem was addressed by using quantification techniques (i.e., Extended Kalman Filters). In [41] the authors survey the methods regarding the exploitation of AIS data towards anomaly detection, including approaches on VCRA from two families of methods, namely Closest Point of Approach (CPA) and Fuzzy Logic. Within the context of CPA-based methods, [5] proposes a unified formulae system that includes not only the distance and time to CPA (DCPA and TCPA, respectively), but also their relative difference in speed and direction, as well as the distance between the two involved vessels.

In contrast to the formulaic approach, [6] proposes a Time Discrete Non-linear Velocity Obstacle (TD-NLVO) method, in which the vessel encounter is considered as a process, rather than analyzing traffic data at certain time slices. Their CRI formula takes into account the vessels’ distance, time difference, as well as their corresponding domain, calculated with respect to their length.

Iphar et al. [15] propose a general-purpose expert-based method for the risk assessment of anomalous maritime transportation data which, among other situations, calculates the vessels’ collision risk using four discrete danger levels. Shi et al. [35] propose a novel track pairs collision detection algorithm for evaluating the risk of vessel collision in port traffic using a variant of Douglas-Peucker algorithm [11], which takes into account the vessels’ speed in order to achieve better compression with respect to quality.

Hongdan et al. [13] propose a deterministic collision risk assessment and avoidance model, based on the vessels’ kinematic characteristics and CPA alarms. Additionally, Lee et al. [17] propose a collision prevention algorithm aimed towards the safety of small fishing vessels. Following this line of research, Zhou et al. [44] propose a collision risk assessment and avoidance model for USV, composed of the vessels’ navigation safety domain and CRI.

Most similar to our work on VCRA, Gang et al. [12] propose an ML-based approach that takes as input the encountering vessels’ CPA-based features and generate a dataset used to train an SVM model. In a similar fashion, Li et al. [18] replace the SVM with a CART model for calculating the vessels’ CRI and assess its performance over five different metrics. In one of the most recent works, Park et al. [29] propose an RVM model for calculating the encountering vessels’ CRI, and assess its performance compared to [12].

2.2 Vessel Collision Risk Forecasting

In relation to the VCRF problem, Sang et al. [33] propose a position prediction model in order to extract short-term future trajectories of vessels using AIS data. In particular, they propose an improvement over the existing CPA calculation methods by adding the change of speed (COS) and rate of turn (ROT), and taking into account all points in the predicted trajectory instead of only the latest transmitted ones. Through the analysis of a real-world collision scenario, they show that the proposed method can help identify and warn of anomalous vessel behaviour in a realistic time frame.

²Interestingly, the collision risk between two vessels is not symmetric; see the related discussion in Section 6

Moving on to DL approaches, Vukša et al [42] propose a vessel collision probability estimation model over a maritime spatial grid based on Monte Carlo simulation and bidirectional long short-term memory neural network (Bi-LSTM). In a similar manner, Liu et al. [20] propose a Convolutional LSTM (ConvLSTM) model, which can extract spatial-temporal features and predict the vessels' collision probability within a region of interest. Further following this line of research, [19] proposes a framework for short-term regional collision risk prediction by combining DBSCAN clustering with Shapley values, a method from game theory, and Recurrent Neural Networks (RNNs). Most recently, Namgung et al. [27] propose a VCRF framework, which consists of a Fuzzy Inference System-based variant (FIS-NC) and an RNN to aid timely decision making.

Most similar to our work on VCRF, Ma et al. [22] reduce VCRF into time-series classification and propose an LSTM-based model which maps the encountering vessels' behavioural features to their corresponding CRI in the future. Furthermore, Ma et al. [21] extend [22] by offering an updated version of the aforementioned model that employs Bi-directional LSTMs [34] and the Attention [3] mechanism to improve feature extraction. Nevertheless, their approach [21,20] is not directly comparable with our framework since they directly predict vessels' CRI instead of relying upon a vessel route prediction model.

3 PROBLEM FORMULATION

In this section, we formulate the VCRA/F problems and present our proposed methodology. The main background definitions are as follows:

Definition 3.1 (Trajectory). A trajectory T_j of vessel j is represented by a sequence of timestamped GPS positions, where the k^{th} time-stamped position is expressed as a triplet of $[t_j^k, lon_j^k, lat_j^k]$, with the values corresponding to timestamp, longitude, and latitude, respectively.

Definition 3.2 (Collision Risk Index). Given the location (lon, lat) , course ϕ , and speed V over ground of two vessels v_O and v_T (for "own" and "target" vessel, respectively), the Collision Risk Index (CRI) of v_O with respect to v_T is expressed as a probability, calculated via the dot-product of Eq. 1, where W denotes the weight vector of target factors, and U denotes the membership vector of target factors, namely the Distance and Time to Closest Point of Approach (DCPA and TCPA, respectively), distance D_O^T , relative bearing θ_O^T , and speed ratio $K = \frac{V_T}{V_O}$ [29].

$$CRI = W \cdot U = W_{DCPA}U_{DCPA} + W_{TCPA}U_{TCPA} + W_{D_O^T}U_{D_O^T} + W_{\theta_O^T}U_{\theta_O^T} + W_KU_K \quad (1)$$

For the purposes of this paper, we adopt the weights and membership functions introduced in [12, 29, 43]; Figure 2 illustrates a diagram of the involved parameters³.

Definition 3.3 (Vessel Encountering Process). Assuming fixed sampling rate⁴, vessels v_O and v_T , are in an encountering process,

³More information on the mathematical formulae can be found at <https://github.com/DataStories-UniPi/VCRA>.

⁴In real-world, AIS signals are not transmitted in fixed sampling; nevertheless, the fixed sampling rate assumption can be realized by, on the one hand, using interpolation

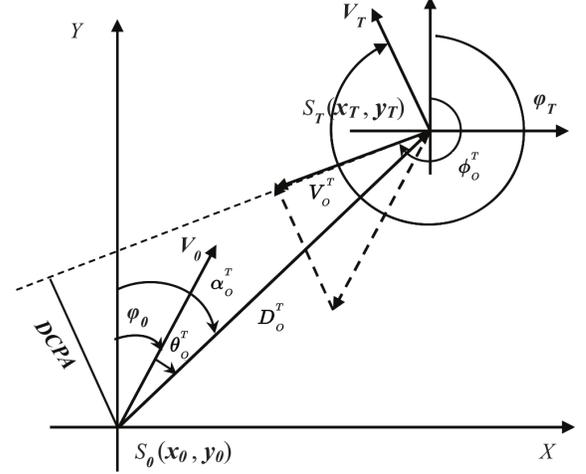


Figure 2: The diagram of vessel collision geometry, adapted from [12].

when their distance D_O^T is less than D_{max} for at least $k > 0$ most recent time-slices and is monotonically decreasing during this period $[t_{now-k}, t_{now}]$.

Definition 3.4 (Vessel Collision Risk Assessment). Given a pair of vessels v_O, v_T that are in an encountering process, VCRA aims to calculate the CRI of v_O with respect to v_T .

Definition 3.5 (Vessel Collision Risk Forecasting). Given a pair of vessels v_O, v_T , and a prediction horizon Δt , VCRF aims to evaluate (i) whether the two vessels will be in an encountering process within Δt based on their anticipated routes and, if yes, (ii) calculate the CRI of v_O with respect to v_T at the time of expected encountering.

Adopting Definition 3.3 with $k = 4$ in the example of Figure 1, we discover encountering process $e_1(a, b)$ at t_{now} , whereas via Definition 3.5 we predict that this encounter will continue for two more time-slices (i.e., e_2 , a superset of e_1), and up to $t_{now} + \Delta t$ we discover a new encountering $e_3(a, c)$.

4 VCRA/F FRAMEWORK OVERVIEW

The proposed architecture of our VCRA/F framework is illustrated in Figure 3. In particular, starting from the offline layer and given the AIS-enabled vessels' historical data, we train a VCRA and a VRF model, the latter being in charge of forecasting future vessel routes (to be used in the VCRF process). Proceeding to the online layer, we feed the preprocessed (cleansed) AIS data-stream to the VRF model in order to get the predicted routes, and for the anticipated encountering vessels' their kinematic information is passed through the VCRA model in order to calculate their CRI.

In contrast to unified frameworks for VCRA/F [22, 21] our proposed framework not only predicts vessels' CRI but also predicts their corresponding locations up to $t_{now} + \Delta t$. This makes our framework modular in terms that, depending on the use-case, different VCRA and/or VRF models can be deployed in a plug and play mode.

in past locations and, on the other hand, setting the VRF model to operate as such when producing future locations.

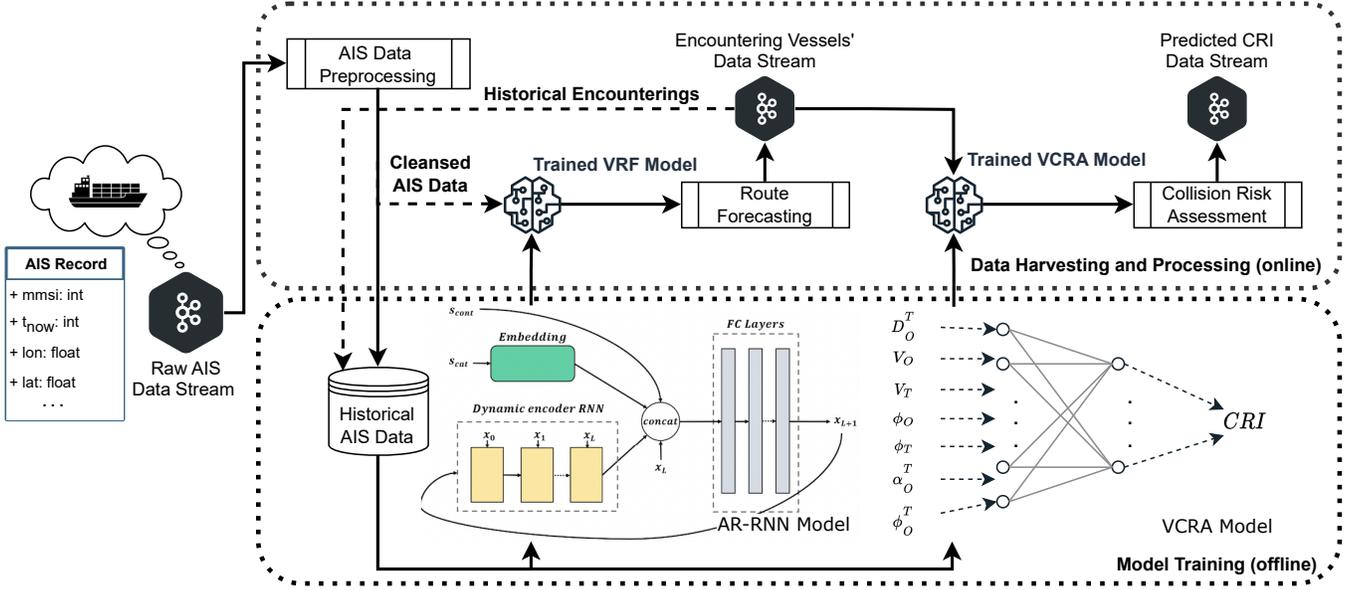


Figure 3: Architecture overview of the proposed VCRA/F framework

In the sections that follow, we present the details of the main components of our architecture, namely, the online layer including the tasks of processing AIS data, tracking encounters, and calculating CRI (Section 4.1) and the offline layer including the underlying VCRA and VRF models (Section 4.2).

4.1 The Online Layer

In this section, we describe the online part of our VCRA/F framework, which is responsible for data preprocessing, tracking encountering vessels, and assessing the CRI of the tracked encountering vessels in the short-term future.

Algorithm 1 illustrates our method for discovering and tracking encountering vessels in online fashion. Firstly, for each incoming time-slice t_i , we retrieve the current vessel pairs $Pairs_{current}$ using the Ball-Tree spatial index provided by scikit-learn ML library⁵ over the vessels' location at t_i (line 3). If there are no actively tracked pairs $Pairs_{active}$, then we directly track $Pairs_{current}$ (lines 4–5). Otherwise, if no pair of vessels are in close vicinity, we stop tracking $Pairs_{active}$ and save/output the actively tracked pairs that satisfy the temporal constraint ($Pairs_{out}$; lines 6–9).

On the other hand, if both $Pairs_{active}$ and $Pairs_{current}$ are non-empty, we update $Pairs_{active}$ using Algorithm 2, with the encountering vessels that no longer maintain distance monotonicity but satisfy the temporal constraint being saved to $Pairs_{inactive}$ (lines 10–13). Finally, the vessels in $Pairs_{active}$ that satisfy the temporal constraint are output in a data-stream for VCRA/F as well as in a data storage for future reference, e.g., model retraining (line 14).

In particular for Algorithm 2, it compares the active ($Pairs_{active}$) with the current ($Pairs_{current}$) encountering pairs of vessels and separates them into the following three cases: an encountering pair

has either disappeared (line 2) or emerged (line 3) or survived (lines 6–16).

For the latter case, if $Pair \in Pairs_{active}$ satisfies the distance monotonicity constraint (lines 13–14) then it is labeled as “survived” and added to the updated $Pairs'_{active}$ set (line 17), whereas if $Pair$ does not meet the aforementioned constraint (lines 10–11), it is labeled as “disappeared” and added to the updated $Pairs'_{inactive}$ set if it satisfies the temporal constraint as well (line 18).

Recalling Figure 1 and using Algorithm 1 for $k = 4$, we discover the encountering process e_1 at time-slice $t_{now} - k$ and track it up to t_{now} , where the corresponding vessels' CRI is calculated. Using a VRF model, we predict all time-slices up to $t_{now} + \Delta t$, where we discover that e_1 continued to be tracked hence, e_2 , with the CRI being calculated as well at each predicted time-slice up to $t_{now} + 2$. Moreover, we discover a new encountering process e_3 at $t_{now} + (\Delta t - 5)$, which continues to be tracked up to $t_{now} + \Delta t$, with the vessels' corresponding CRI being calculated at the last two time-slices ($t \in [t_{now} + (\Delta t - 1), t_{now} + \Delta t]$) due to $k = 4$.

4.2 The Offline Layer

For the VCRA task, we employ a variant of the MLP-VCRA model proposed in [37]. Essentially, we use a Multi-Layered Perceptron (MLP) model, with two hidden layers of 256 and 32 neurons each (chosen empirically), trained for 100 epochs using the Adam [16] optimizer and early stopping [32] in order to avoid overfitting.

After calculating the actual CRI of “own” with respect to “target” vessel (to be used as ground truth), we create a dataset with seven features, namely D_O^T , V_O , V_T , ϕ_O , ϕ_T , α_O^T and ϕ_O^T (c.f., Figure 2), and train the aforementioned model in order to assess vessels' CRI, without resorting to computationally complicated and “expensive” formulae.

⁵<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.BallTree.html>

Algorithm 1: VESSELENCOUNTERS. Online identification of vessels in encountering process within temporally aligned trajectory data-streams.

Input: Data-stream $D = \{t_1, t_2, \dots, t_n\}$ of time-slices t_i consisting of vessels' time-stamped locations; Distance threshold D_{max} ; Temporal threshold k ;

Output: Vessels' pairs that satisfy Definition 3.3

```

1  $Pairs_{active}, Pairs_{inactive} \leftarrow \emptyset$ 
2 for time-slice  $t_i \in D$  do
3    $Pairs_{current} \leftarrow \text{CURRENTPAIRS}(t_i, D_{max})$ 
4   if  $Pairs_{active} == \emptyset$  then
5      $Pairs_{active} \leftarrow Pairs_{current}$ 
6   else if  $Pairs_{current} == \emptyset$  then
7      $Pairs_{out} = \{Pair \in Pairs_{active} : Pair.end - Pair.start \geq k\}$ 
8      $Pairs_{inactive} \leftarrow Pairs_{inactive} \cup Pairs_{out}$ 
9      $Pairs_{active} \leftarrow \emptyset$ 
10  else
11     $Pairs_{active}, Pairs'_{inactive} \leftarrow \text{UPDATEPAIRS}(Pairs_{active}, Pairs_{current}, k)$ 
12     $Pairs_{inactive} \leftarrow Pairs_{inactive} \cup Pairs'_{inactive}$ 
13     $Pairs_{out} = \{Pair \in Pairs_{active} : Pair.end - Pair.start \geq k\}$ 
14  output  $Pairs_{out}$ 
15 end
16 return  $Pairs_{active}, Pairs_{inactive}$ 

```

For predicting vessels' locations in short-term, we use the Auto-Regressive Recurrent Neural Network (AR-RNN) model proposed in [26], which operates as follows. For a given region of interest, unique trips with common origins and destinations are grouped. Each trip (i.e., distinct origin-destination combination) constitutes the training data for a specific RNN model; therefore, there exist as many RNN models as the distinct groups of trips. Given the known origin and destination of a vessel, the respective model is used to forecast its route. Alternatively, a single model can be trained on all underlying data for the region if no such information is available, albeit with lower fidelity compared to decomposing the vessel behavior into specific models [26].

To forecast the route of a vessel v_i , its current trajectory up to t_{now} (c.f., Figure 1) is fed into an RNN in order to produce an embedding $T_{v_i}^{embed}$ of its current mobile status. Afterwards, we concatenate it with another embedding vector produced using the static data of the corresponding vessel (length, etc.). Thus, similar vessels should generate similar embeddings that can be used for more accurate route predictions. Additionally, since the next location of a vessel is highly dependent on the previous, its latest known position, $\{T_{v_i}\}^{t_{now}}$, is concatenated with the aforementioned embeddings, so as to allow the model to learn short-term dependencies.

In this study, the RNN architecture utilized is a Gated Recurrent Unit (GRU) [7] with a hidden size of 50, chosen empirically. This architecture addresses the issue of vanishing gradients in "vanilla" RNNs, whilst reducing the number of trainable parameters compared to the more commonly used LSTM architecture.

Algorithm 2: UPDATEPAIRS. Compares active with the current vessel pairs in order to determine their evolution.

Input: Active vessel pairs $Pairs_{active}$; Current vessel pairs $Pairs_{current}$; Temporal threshold k

Output: Updated active vessel pairs $Pairs'_{active}$; Inactive vessel pairs $Pairs'_{inactive}$

```

1  $survived \leftarrow \emptyset$ 
2  $disappeared \leftarrow \{Pair \in Pairs_{active} : Pair \notin Pairs_{current}\}$ 
3  $emerged \leftarrow \{Pair \in Pairs_{current} : Pair \notin Pairs_{active}\}$ 
4  $Pairs_{active} \leftarrow Pairs_{active} - disappeared$ 
5  $Pairs_{current} \leftarrow Pairs_{current} - emerged$ 
6 for  $Pair_{active} \in Pairs_{active}$  do
7   for  $Pair_{current} \in Pairs_{current}$  do
8     if  $Pair_{active}.pair \neq Pair_{current}.pair$  then
9       continue
10    else if  $Pair_{active}.distance < Pair_{current}.distance$  then
11       $disappeared \leftarrow disappeared \cup Pair_{active}$ 
12    else
13       $Pair_{current}.start \leftarrow Pair_{active}.start$ 
14       $survived \leftarrow survived \cup Pair_{current}$ 
15    end
16 end
17  $Pairs'_{active} \leftarrow emerged \cup survived$ 
18  $Pairs'_{inactive} \leftarrow \{Pair \in disappeared : Pair.end - Pair.start \geq k\}$ 
19 return  $Pairs'_{active}, Pairs'_{inactive}$ 

```

The resultant vector is fed into a 2-layer regression head of fully connected (FC) layers with a hidden layer size of 27, and output layer of size 2, all empirically selected. The network is trained to predict the next position, $[t_j^{now+1}, lon_j^{now+1}, lat_j^{now+1}]$. This is then fed back into the network in an auto-regressive manner, until the desired prediction horizon is reached. The network is trained using the Adam [16] optimizer and early stopping [32] to avoid overfitting.

5 EXPERIMENTAL STUDY

In this section, we evaluate the proposed VCRA/F model using a real-world AIS dataset in comparison with related work [12, 18, 29] that has been reproduced for the purposes of our experimental study.

5.1 Experimental Setup

Our VCRA/F framework is implemented in Python. For our experimental study, we use a GPU cluster owned by the University of Piraeus, out of which we used 1 Nvidia A100 GPU, 8 CPUs, and 1TB of RAM. The corresponding source-code used in our experiments is available at: <https://github.com/DataStories-UniPi/VCRA>.

For the purpose of our experimental study, we use a subset of the "Historical AIS data in Norwegian waters" [10], hereafter referred to as the "Norway"⁶ dataset, which consists of AIS records transmitted by 732 distinct vessels in Jan. 2019; Figure 4 illustrates a single-day snapshot of it. In terms of preprocessing, because

⁶The dataset is publicly available at <https://ais-public.kystverket.no/>



Figure 4: A snapshot of the Norway dataset on Jan. 10th, 2019.

#Records	8,352,352
#Vessels	732
#Segments	7267
#Points per Segment (<i>min</i> ; <i>med.</i> ; <i>avg.</i> ; <i>max.</i>)	20; 172; 1153; 77759
Vessels' Speed (<i>min</i> ; <i>avg.</i> ; <i>max.</i>)	0; 3; 50 knots
Sampling Rate	30 sec.

Table 1: Statistics of Norway dataset after preprocessing

GPS/AIS data is prone to noisy and/or erroneous records [30], we drop the records with speed above 50 knots as well as the ones with speed below 1 knot (considered stationary points). Furthermore, we segment the vessels' locations to port-by-port trajectories, with an additional segmentation when a pair of points with a temporal difference higher than 30 minutes is detected. Finally, we use linear interpolation with a sampling rate of 30 seconds to temporally align the vessels' locations; Table 1 illustrates the dataset statistics after the aforementioned preprocessing task. Regarding the underlying models, the VCRA model is trained over this dataset, as discussed in the following section, whereas the model used for VRF purposes is a pre-trained model, as presented in [26].

5.2 Experimental Results on VCRA

In this section, we evaluate the VCRA part of our framework. After running Algorithm 1 on the aforementioned dataset for $D_{max} = 1$ n.m. and $k = 2$ time-slices (i.e., 1 min. using a sampling rate of 30 sec.), we detect 62,359 vessel encounters with 359,846 training samples on encountering vessels' CRI. Table 2 presents the performance of our framework in what regards its VCRA part, compared to related work in terms of Mean Absolute Error (MAE), Root Means

Squared [Log] Error (RMS[L]E), and R^2 score, in relation to CRI determined using mathematical formulae. Due to the spatial complexity of RVM-VCRA [29] we use a stratified subset that consists of 125,946 records (35%), which is split into training and test sets using a stratified 5-fold cross-validation method. It can be observed that our approach clearly outperforms its competitors by a high margin (up to 70%) in terms of R^2 score, while reaching an overall accuracy of about 96%. Moreover, the comparatively low RMS[L]E leads us to the conclusion that our model has a lower tendency to underestimate danger (in terms of CRI), resulting in fewer false negatives, which in our case is more favourable than a false positive alert within the context of ensuring maritime traffic safety.

	MAE	RMSE	RMSLE	R^2
Gang et al. [12]	0.1194	0.1969	0.1452	0.5766
Li et al. [18]	0.0395	0.1165	0.0853	0.8517
Park et al. [29]	0.1272	0.1936	0.1379	0.5906
VCRA/F	0.0246	0.0607	0.0440	0.9597

Table 2: Comparing our VCRA approach with respect to related work in terms of MAE and RMS[L]E error (the lower the better) and R^2 score (the higher the better).

After training the top performing models (i.e., Li et al. [18], and our VCRA/F) on the full training dataset using stratified 5-fold cross-validation, Table 3 presents the models' predictive behaviour, in terms of RMSLE, across five risk levels. It can be observed that our model has more "confident" predictions on low (< 0.2) and (medium-) high (> 0.6) CRI, while low-medium risk encounters ($0.2 < CRI \leq 0.4$) have a tendency to be over-estimated, mostly concentrated at medium risk level ($0.4 < CRI \leq 0.6$). This behaviour may be attributed to the (natural) imbalance of the dataset regarding the CRI (most encountering processes have either zero or high risk), as Figure 5 illustrates.

	[0, 0.2]	(0.2, 0.4]	(0.4, 0.6]	(0.6, 0.8]	(0.8, 1]
Li et al. [18]	0.1795	0.0663	0.0675	0.0585	0.0389
VCRA/F	0.0869	0.0760	0.0496	0.0312	0.0215

Table 3: Comparing the predictive behaviour of our VCRA approach with respect to related work in terms of RMSE across five distinct CRI risk levels.

Additionally, Table 4 presents the response time, i.e., the latency of calculating a CRI value using the "busiest" time-slice (in terms of concurrent encounterings) of the Norway dataset ($n = 40$). It can clearly be observed that our approach not only outperforms all aforementioned works by a significant margin, but also the CRI formula (c.f., Eq. 1) used to create the ground truth.

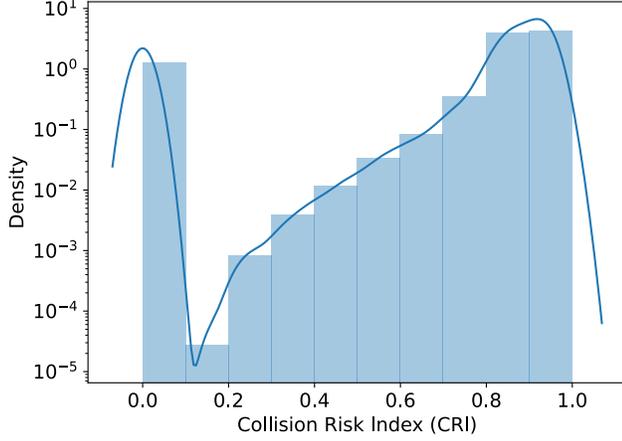


Figure 5: Probability density function of the training dataset

Method	Latency ($\mu \pm \sigma$)
CRI Formula (Eq. 1)	2.30 msec. $\pm 47\mu\text{sec.}$
Gang et al. [12]	43.4 msec. $\pm 430\mu\text{sec.}$
Li et al. [18]	3.38 msec. $\pm 56\mu\text{sec.}$
Park et al. [29]	0.29 msec. $\pm 45\mu\text{sec.}$
VCRA/F	0.22 msec. $\pm 110\mu\text{sec.}$

Table 4: Comparing the response time of the different VCRA models.

5.3 Experimental Results on VCRF

The evaluation of a VCRF approach is not a straightforward task, since we need to define how the error between the predicted and the actual encountering process will be quantified. Intuitively, our aim is to match each encounter EP_{pred} with the most similar actual cluster EP_{act} . This matching will yield encounters that are predicted accurately (i.e., true positives) as well as actual encounters that were not predicted (i.e., false negatives) and predicted encounters that were not discovered in the actual set (i.e., false positives). Towards evaluating the true positive encounterings, we adopt Allen’s interval algebra [1] and calculate their similarity as follows:

$$Sim(EP_{pred}, EP_{act}) = \frac{Interval(EP_{pred}) \cap Interval(EP_{act})}{Interval(EP_{pred}) \cup Interval(EP_{act})} \quad (2)$$

where $Interval(EP_{pred})$, ($Interval(EP_{act})$, respectively) is the time interval when the predicted (actual, respectively) encountering process was valid.

Based on the above, we assess the quality of the VCRF part of our framework with respect to its corresponding “ground truth”. As “ground truth” we define the encountering vessels discovered using the objects’ corresponding actual positions and assessed using Eq. 1, and for quantifying the accuracy of a predicted encounter EP_{pred} against an actual one EP_{act} , we use Eq. 2.

Focusing on the encountering cargo/passenger vessels inside a specific region of interest (Oslo Fjord), we discover 36 actual encountering processes, while we predict 26 encountering processes using the underlying VRF and VCRA models. Out of these, we have a match (i.e., same vessels) at 24 encounters, meaning that we have 2 false positives (i.e., predicted encounters that did not actually happen) as well as 12 false negatives (i.e., actual encounters that were not discovered). Furthermore, for the matched encounters, using Eq. 2, we observe that our VCRF framework has accurately predicted them with $\approx 77\%$ accuracy, in terms of temporal overlap.

To illustrate the above discussion, Figure 6 presents high-risk (in terms of CRI) areas of interest on actual vs. predicted locations, respectively. We observe that for the actual regions (c.f., Figure 6a) most encounters are concentrated near the “entrance” of the Fjord, with some moderate risk areas, which may indicate either vessel anchorages [40], or high flow maritime “highways” that vessels may use for entering/exiting the Fjord.

On the other hand, for the predicted high-risk areas of interest (c.f., Figure 6), we observe that we accurately predict most vessel encounters within the Fjord, albeit with slightly lower mean CRI on the Fjord entrance, as well as nearby ports. For the case of the aforementioned maritime “highways”, our predictions yield less high-risk regions, with slightly less mean CRI. These findings may trigger domain experts into further investigating these occurrences and reach meaningful conclusions.

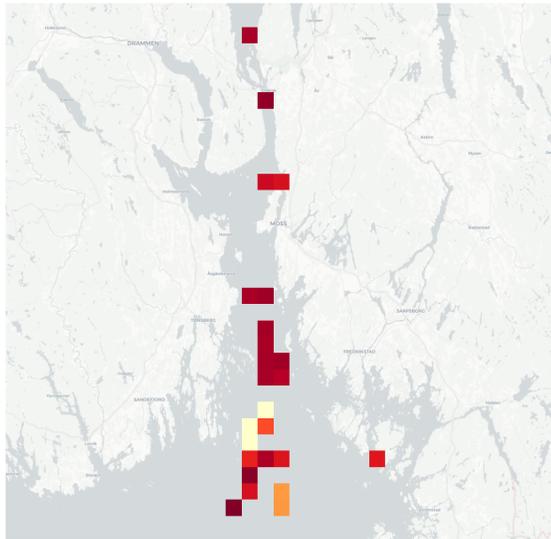
For forecasting future encountering processes within a $\Delta t = 50$ minute prediction horizon, a single VRF prediction takes only 0.86 msec. for the aforementioned time-slice, resulting in a total latency of 1.2 msec. on average, rendering our VCRA/F framework a feasible solution for online streaming maritime safety systems.

6 A NOTE ON MODEL TRANSPARENCY

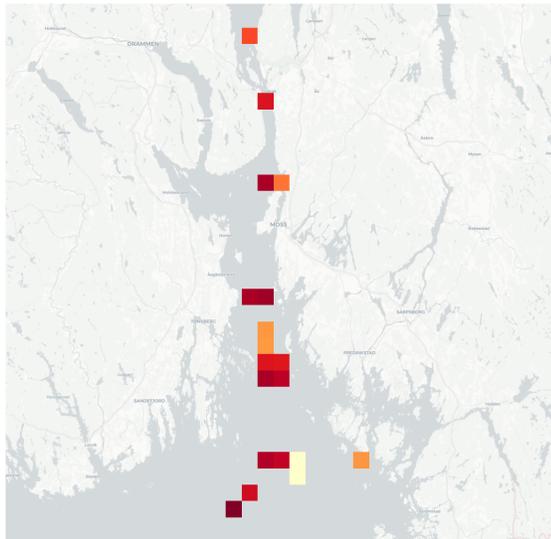
While ML/DL-based models are capable of providing state-of-the-art results, in many cases they are used (and regarded) as black-boxes, with little-to-no knowledge on the rationale behind their decision. SHapley Additive exPlanations (SHAP) is a popular technique in the field of eXplainable AI (XAI) that seeks to provide insights into the decision-making process of models. It specifically focuses on explaining individual predictions by utilizing the game theoretically optimal Shapley values. Shapley values, derived from cooperative game theory, possess favorable characteristics and are widely adopted. In this context, the values of the features in a given data instance are treated as players within a coalition. The Shapley value represents the average marginal contribution of a feature value considering all potential coalitions [24].

Figure 7 illustrates the summary of SHAP values for each feature of a random subset of the test set of our VCRA model. At first glance, it is clear that the vessels’ speed (V_O , V_T) and direction (ϕ_O , ϕ_T) have the least impact, whereas their distance (D_O^T), azimuth angle (α_O^T), and relative movement direction (ϕ_O^T) have the most significant impact, as they determine not only how close the “own” is to the “target” vessel, but also their relative positioning, in accordance with the vessel collision regulations and providence measures, as illustrated in Figure 10.

In a more thorough view, we observe that as D_O^T increases, the more negative impact (i.e., it contributes to the decrease of the



(a)



(b)



Figure 6: Visualization of (a) actual vs. (b) predicted high-risk (in terms of CRI) areas of interest within the Oslo Fjord, defined by the window [10.161, 11.077] lon, [58.670, 59.911] lat, with each cell of dimension 6 x 6 km²

output) it has on the CRI calculation. In conjunction with Figure 8, it can be observed that for $D_O^T > 0.7$ the SHAP value for D_O^T decreases rapidly with negative impact on the model output, whereas for

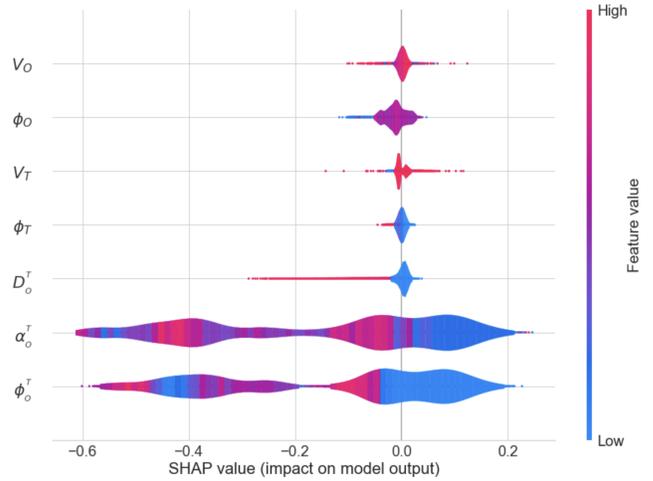


Figure 7: Impact of VCRA input variables on the model output

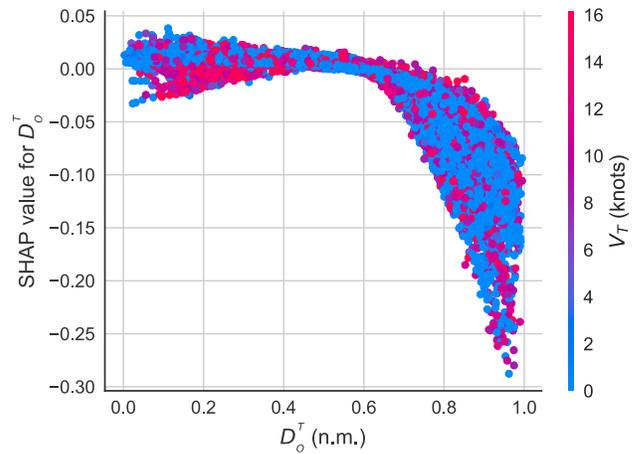


Figure 8: Impact of D_O^T vs. V_T with respect to the VCRA model output

$D_O^T \leq 0.7$ its impact is positive (i.e., it contributes to the increase of the output), albeit minor, when compared to α_O^T and ϕ_O^T .

In contrast to D_O^T , assessing the impact of α_O^T and ϕ_O^T on their own is quite cumbersome, since they display different values on negative SHAP values. However, when jointly observing α_O^T and ϕ_O^T , it seems that they are correlated, since they display a similar spectrum of feature values across negative SHAP values. Looking at Figure 9, we can clearly observe that when $\phi_O^T \leq \pi$, $\alpha_O^T \geq \pi$ with positive SHAP values (i.e., increasing CRI), whereas when $\phi_O^T > \pi$, $\alpha_O^T \geq \pi$ with negative SHAP values (i.e., decreasing CRI). This conclusion is compliant with the stand-on/give-way rules in maritime (c.f., Figure 10).

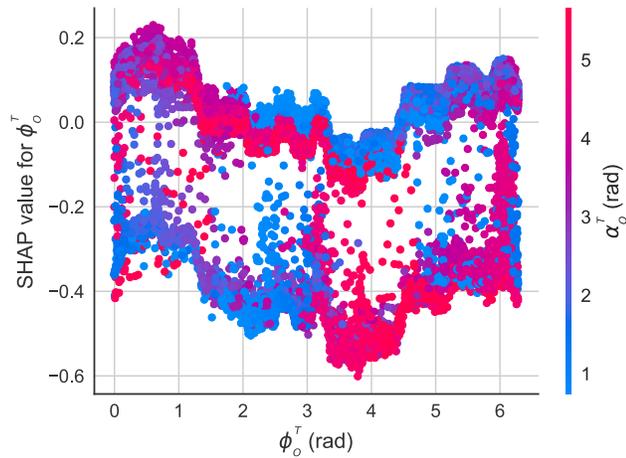


Figure 9: Impact of ϕ_o^T vs. α_o^T (with respect to the VCRA model output)

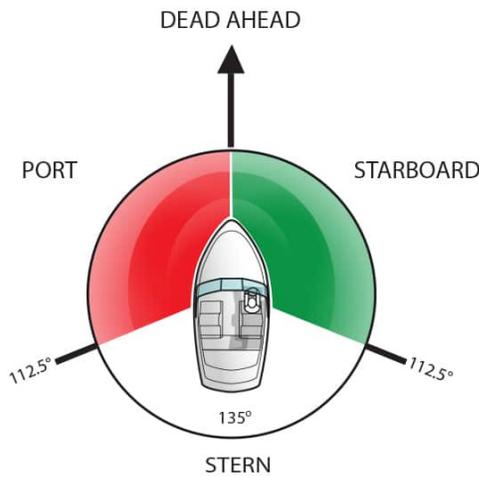


Figure 10: Vessel collision regulations - determining right-of-way (source: <https://www.boatsmartexam.com>).

7 CONCLUSION

In summary, in this paper we studied the VCRA/F problem from the ML perspective and proposed an efficient ML-based solution, which consists of an online layer that is responsible for AIS data preprocessing, vessel encountering tracking, and CRI calculation on encountering vessels, and an offline layer that maintains two effective models, for the purposes of VCRA and VRF, respectively. Our experimental results on a real-world AIS dataset demonstrate the efficiency of the proposed model compared to related work, in terms of both quality and latency.

In the near future, we aim to further demonstrate the efficacy of VCRA in real-world situations through Visual Analytics frameworks [39] as well as experiment on advanced VRF models [8, 9] in order to evaluate whether the accuracy of VCRA/F can be further

improved. Moreover, we intend to further advance our VCRA/F solution through state-of-the-art collision avoidance methods [14] so as to be able to address the collision avoidance problem as well. As a long-term goal, we aim to experiment on the parameter sensitivity of VCRA/F with respect to its hyperparameters (i.e., D_{max} , k) as well as its prediction capabilities in terms of sparse vessel trajectories. Finally, regarding the offline part of models' training, we aim to exploit on Federated Learning [23] in order to train our solution while preserving the vessel owners' privacy, as well as Lifelong Learning [28] in order to facilitate gradual model improvement over time in terms of incorporating new information.

ACKNOWLEDGMENTS

This work was supported in part by EU Horizon 2020 R&I Programme under Grant Agreement No 957237 (project VesselAI, <https://vessel-ai.eu>).

REFERENCES

- [1] James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26, 11, 832–843.
- [2] Alexander Artikis and Dimitris Zissis. 2021. *Guide to Maritime Informatics*. Springer New York, 1–333. ISBN: 978-3-030-61852-0.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
- [4] Sable Campbell, Wasif Naeem, and George W. Irwin. 2012. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36, 2, 267–283.
- [5] Dejun Chen, Chu Dai, Xuechao Wan, and Junmin Mou. 2015. A research on ais-based embedded system for ship collision avoidance. In *Proceedings of the 3rd International Conference on Transportation Information and Safety (ICTIS)*, 512–517.
- [6] Pengfei Chen, Yamin Huang, Junmin Mou, and PHAJM van Gelder. 2018. Ship collision candidate detection method: a velocity obstacle approach. *Ocean Engineering*, 170, 186–198.
- [7] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: encoder-decoder approaches. In *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)*.
- [8] Eva Chondrodima, Petros Mandalis, Nikos Pelekis, and Yannis Theodoridis. 2022. Machine learning models for vessel route forecasting: an experimental comparison. In *Proceedings of the 23rd IEEE International Conference on Mobile Data Management (MDM)*, 262–269.
- [9] Eva Chondrodima, Nikos Pelekis, Aggelos Pikrakis, and Yannis Theodoridis. 2023. An efficient LSTM neural network-based framework for vessel location forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 24, 5, 4872–4888.
- [10] Julien Desjardins. [n. d.] Historical AIS data in norwegian waters. Last Visited: 2022/02/01. <https://ais-public.kystverket.no/ais-download>.
- [11] David H. Douglas and Thomas K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10, 112–122.
- [12] Longhui Gang, Yonghui Wang, Yao Sun, Liping Zhou, and Mingheng Zhang. 2016. Estimation of vessel collision risk index based on support vector machine. *Advances in Mechanical Engineering*, 8, 11, 1687814016671250.
- [13] Liu Hongdan, Liu Qi, and Sun Rong. 2019. Deterministic vessel automatic collision avoidance strategy evaluation modeling. *Intelligent Automation & Soft Computing*, 25, 4, 789–804.
- [14] Yamin Huang, Linying Chen, Pengfei Chen, Rudy R. Negenborn, and P.H.A.J.M. van Gelder. 2020. Ship collision avoidance methods: state-of-the-art. *Safety Science*, 121, 451–473.
- [15] Clément Iphar, Aldo Napoli, and Cyril Ray. 2020. An expert-based method for the risk assessment of anomalous maritime transportation data. *Applied Ocean Research*, 104, 102337.
- [16] P. D. Kingma and J. Ba. 2015. Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.
- [17] Myoung-Ki Lee and Young-Soo Park. 2020. Collision prevention algorithm for fishing vessels using mmwave communication. *Journal of Marine Science and Engineering*, 8, 2.

- [18] Yishan Li, Zhiqiang Guo, Jie Yang, Hui Fang, and Yongwu Hu. 2018. Prediction of ship collision risk based on CART. *IET Intelligent Transport Systems*, 12, 10, 1345–1350.
- [19] Dapei Liu, Xin Wang, Yao Cai, Zihao Liu, and Zheng-Jiang Liu. 2020. A novel framework of real-time regional collision risk prediction based on the RNN approach. *Journal of Marine Science and Engineering*, 8, 3.
- [20] Ryan Wen Liu, Xiaojie Huo, Maohan Liang, and Kai Wang. 2022. Ship collision risk analysis: modeling, visualization and prediction. *Ocean Engineering*, 266.
- [21] Jie Ma, Chengfeng Jia, Xin Yang, Xiaochun Cheng, Wenkai Li, and Chunwei Zhang. 2020. A data-driven approach for collision risk early warning in vessel encounter situations using attention-bilstm. *IEEE Access*, 8, 188771–188783.
- [22] Jie Ma, Wenkai Li, Chengfeng Jia, Chunwei Zhang, and Yu Zhang. 2020. Risk prediction for ship encounter situation awareness using long short-term memory based deep learning on intership behaviors. *Journal of Advanced Transportation*, 2020.
- [23] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629.
- [24] Christoph Molnar. 2022. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. (2nd ed.). <https://christophm.github.io/interpretable-ml-book>.
- [25] Brian Murray and Lokukaluge Prasad Perera. 2020. A dual linear autoencoder approach for vessel trajectory prediction using historical ais data. *Ocean Engineering*, 209, 107478.
- [26] Brian Murray and Lokukaluge Prasad Perera. 2021. An AIS-based deep learning framework for regional ship behavior prediction. *Reliability Engineering & System Safety*, 215, 107819.
- [27] Ho Namgung and Sung-Wook Ohn. 2022. Fuzzy inference and sequence model-based collision risk prediction system for stand-on vessel. *Sensors*, 22, 13.
- [28] German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 54–71.
- [29] Jinwan Park and Jung-Sik Jeong. 2021. An estimation of ship collision risk based on relevance vector machine. *Journal of Marine Science and Engineering*, 9, 5.
- [30] Nikos Pelekis and Yannis Theodoridis. 2014. *Mobility Data Management and Exploration*. Springer.
- [31] Lokukaluge P Perera and C Guedes Soares. 2015. Collision risk detection and quantification in ship navigation with integrated bridge systems. *Ocean Engineering*, 109, 344–354.
- [32] Lutz Prechelt. 2012. Early stopping - but when? In *Neural Networks: Tricks of the Trade (2nd ed.)* Lecture Notes in Computer Science. Vol. 7700. Springer, 53–67.
- [33] Ling-zhi Sang, Xin-ping Yan, Alan Wall, Jin Wang, and Zhe Mao. 2016. CPA calculation method based on ais position prediction. *Journal of Navigation*, 69, 6, 1409–1426.
- [34] M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45, 11, 2673–2681.
- [35] Jiahui Shi and Zhengjiang Liu. 2022. Track pairs collision detection with applications to ship collision risk assessment. *Journal of Marine Science and Engineering*, 10, 2.
- [36] Panagiotis Tampakis, Eva Chondrodima, Andreas Tritsarolis, Aggelos Pikrakis, Yannis Theodoridis, Kostis Pristouris, Harry Nakos, Panagiotis Kalampokis, and Theodore Dalamagas. 2022. I4sea: a big data platform for sea area monitoring and analysis of fishing vessels activity. *Geo-spatial Information Science*, 25, 2, 132–154.
- [37] Andreas Tritsarolis, Eva Chondrodima, Nikos Pelekis, and Yannis Theodoridis. [n. d.] Vessel collision risk assessment using AIS data: A machine learning approach. In *Proceedings of the 23rd IEEE International Conference on Mobile Data Management (MDM)*, 425–430.
- [38] Andreas Tritsarolis, Eva Chondrodima, Panagiotis Tampakis, Aggelos Pikrakis, and Yannis Theodoridis. 2022. Predicting co-movement patterns in mobility data. *Geoinformatica, online first*.
- [39] Andreas Tritsarolis, Christos Doukeridis, Nikos Pelekis, and Yannis Theodoridis. 2021. ST_Visions: a python library for interactive visualization of spatio-temporal data. In *Proceedings of the 22nd IEEE International Conference on Mobile Data Management (MDM)*, 244–247.
- [40] Andreas Tritsarolis, Yannis Kontoulis, Nikos Pelekis, and Yannis Theodoridis. 2021. MaSEC: discovering anchorages and co-movement patterns on streaming vessel trajectories. In *Proceedings of the 17th International Symposium on Spatial and Temporal Databases (SSTD)*, 170–173.
- [41] Enmei Tu, Guanghao Zhang, Lily Rachmawati, Eshan Rajabally, and Guang-Bin Huang. 2018. Exploiting ais data for intelligent maritime navigation: a comprehensive survey from data to methodology. *IEEE Transactions on Intelligent Transportation Systems*, 19, 5, 1559–1582.
- [42] Srđan Vukša, Pero Vidan, Mihaela Bukljaš, and Stjepan Pavić. 2022. Research on ship collision probability model based on monte carlo simulation and bi-lstm. *Journal of Marine Science and Engineering*, 10, 8.
- [43] Shaobo Wang, Yingjun Zhang, and Yisong Zheng. 2021. Multi-ship encounter situation adaptive understanding by individual navigation intention inference. *Ocean Engineering*, 237, 109612.
- [44] Jian Zhou, Feng Ding, Jiakuan Yang, Zhengqiang Pei, Chenxu Wang, and Anmin Zhang. 2021. Navigation safety domain and collision risk index for decision support of collision avoidance of USVs. *International Journal of Naval Architecture and Ocean Engineering*, 13, 340–350.