Learning Representative Vessel Trajectories Using Behavioral Cloning

Jan Löwenstrom*, Edgardo Solano-Carrillo†, and Jannis Stoppe‡
Institute for the Protection of Maritime Infrastructures
German Aerospace Center (DLR)
Fischkai 1, 27572 Bremerhaven, Germany
*jan.loewenstrom@dlr.de, †edgardo.solanocarrillo@dlr.de,‡jannis.stoppe@dlr.de

***Abstract*—We suggest a data-driven approach to predict vessel trajectories by mimicking the underlying policy of human captains. Decisions made by those experts are recorded by the automatic identification system (AIS) signals and can be fused with additional non-kinematic factors like destination, weather condition, current tide level or ship size to get a more accurate snapshot of the situation that led to chosen maneuvers. In this work, we explore the usage of a method meant for optimal control, namely Behavioral Cloning, in a forecasting problem, in order to generate end-to-end vessel trajectories purely based on a given initial state. The training and test datasets consist of trajectories from the coast of Bremerhaven, having more than one thousand unique ships and different motion clusters. These are processed by a single deep-learning model, showing promising results in terms of accuracy and providing a research avenue for a near real-time application where vessel trajectories are to be forecast from a given snapshot of the situation — not from the costly history of all the vessels present.**

***Index Terms*—behavioral cloning, vessel trajectory prediction, imitation, reinforcement learning, maritime situational awareness**

I. Introduction

Including a robust model that is capable of predicting accurate future vessel trajectories into any maritime surveillance system is advantageous, as it improves the overall situational awareness. In this context, the ability to generate end-to-end trajectories can be utilized in a variety of different tasks, e.g. detecting anomalous ship behavior or collision avoidance.

However, building such a sophisticated model is challenging. Although the automatic identification system (AIS) made it possible to collect massive amounts of historical ship records — like positions, identities, and other voyage-related information — in order to construct a data-driven approach, it still has some limitations. For example, the highly irregular time sampling or poor data quality, as given by the inaccuracy of the received position of ~ 10 meters [1], or the unreliability of some of the provided information [2].

In addition, real-world ship maneuvering is heavily influenced by external factors like weather conditions, sea currents, tides and surrounding ships [3]. The proposed model has to be aware of this wide range of non-kinematic elements complementary to the movement indicators (current position, speed, heading) in order to capture the underlying causalities of decisions that led to the resulting trajectories. Besides the high-dimensional and semantically varying feature space, a suitable system is required to have the ability of predicting future vessel trajectories for a time horizon of up to an hour or more while retaining low computational costs, with the goal of potentially providing near real-time predictions.

We approach this task from the perspective of imitation learning, where an artificial agent tries to extract and mimic the behavior of human captains in order to generate similar vessel trajectories. One major assumption is that, by learning from huge amounts of historical expert decisions under certain environmental situations, the agent acquires the skill to generalize well enough to assemble tracks based on states it did not encounter before as well. Here, we do also make the assumption that the generalization and average strategy of multiple human experts is based on navigational rules which are given to everyone, such as COLREG [4] or the harbor specific rules. In this regard, the proposed system and model are trained on a specific region of interest to capture the local conditions without the intent to transfer knowledge or generalize to any other coast or harbor.

Being in the realm of Reinforcement Learning — which leverages from an artificial decision maker — allows us to have a flexible prediction scheme without any restriction to a fixed-sized prediction horizon, unlike most of the predominant approaches for the trajectory prediction problem using sequence to sequence models.

II. Related Work

The topic of predicting vessel trajectories has been approached in a variety of different ways, which can be classified into model-driven and data-driven methods. In a model-driven scenario, future trajectories are calculated based upon a realistic modeling of the sea environment, derived purely from the vessel's current position and sailing velocity without learning knowledge from historical traffic data [5]. To this extent, prominent methods in the trajectory prediction literature are the constant velocity model [6], nearly constant velocity model [7], Gaussian process model [8] or Ornstein-Uhlenbeck stochastic process [9].

As mentioned in section I, real-world ship maneuvering is a complex subject that depends on underlying semantics that are hard to incorporate into a model, especially for the fact that the areas of interest differ from one another in terms of e.g. waterways or other region-related rules. Data-driven approaches try to overcome this challenge by learning from historical traffic patterns and thus implicitly extracting representative behavior. Here, works with machine learning methods based on the extended Kalman filter [10], random forest [11], support vector machines [12] or particle filter [13] have been proposed.

In the past years, deep learning approaches based on recurrent neural networks such as the long-short term memory (LSTM) [14], as well as encoder-decoder models, have become the most dominate approach towards sequence-to-sequence tasks and trajectory prediction [5], [15], [16]. In [17], an approach called FRA-LSTM is proposed, which uses two components: a forward sub-network that combines an LSTM with an attention mechanism to mine features of historical AIS trajectory data; and another sub-network that combines a bidirectional LSTM and attention mechanism to mine features of backward historical trajectory data. In the end, the output features extracted from the forward and reverse sub-networks are fused in order to generate the final trajectory prediction. The authors exclusively use kinematic factors and time for the feature space.

A different approach is presented by [18], who use a geometrical similarity-driven method. They exploit the Dynamic Time Warping algorithm to find the most similar trajectory, calculate the distances of the current point as well as the last three points (each 10 minutes apart from each other) and use those four distances as input for an LSTM to predict the next four distances in order to generate the target trajectory of up to 40 minutes. The location prediction error for a horizon of 10 minutes into the future is 0.390 km, and 1.569 km for a horizon of 40 minutes.

In general, works that use sequence to sequence models (like LSTM) for sequence prediction [19]–[23], are limited to a fixed input and output windows (i.e. they map a fixed-size rolling history of past observations to a fixed-size horizon of future predictions). Experience shows that the size of these two rolling windows should be chosen to be close to each other for non-seasonal time series [24] — for seasonal series, the input window should cover at least a full seasonal period. This has two main disadvantages: 1) for long prediction horizons, the networks require long input windows, which are hard to process since their forgetting mechanisms result in exponential decay of information. Recent efforts to remedy this include enforcing power law forget gates [25]; 2) these methods are not scalable. For near real-time applications, where accurate long-term predictions of potentially many future vessels trajectories are needed (say, to reason about collision avoidance), the processing system would need to keep a large buffer of input windows per vessel. We explore here the prediction of complete trajectories to a destination from single initial states.

III. Approach

We consider the task of vessel trajectory forecasting as a component for enhancing the situational awareness of maritime surveillance centers, as illustrated in Fig. 1. The figure

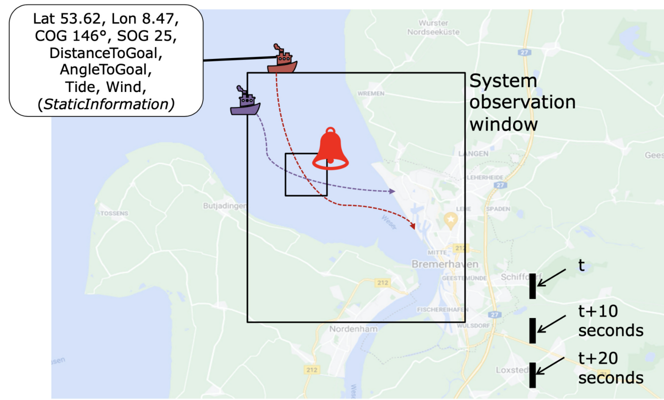


Fig. 1. System Observation Window

shows a scenario where a ship (red color) enters a predefined *system observation window* (SOW). Unlike the sequence to sequence methods in Sec. II, requiring a past sequence of states of size comparable to the prediction horizon, our proposed system is aimed at generating a prediction of the whole trajectory purely based on the initial state. At this moment, the system receives an AIS message with position, COG and SOG; it constructs the full state by fusing destination, tide, and weather information in order to forecast the complete trajectory. The length of the output is flexible and predicted points have a fixed time interval of 10 seconds. If another ship (blue color) enters the SOW, another trajectory will be predicted, allowing for a granular check for close encounters to eventual yield an alarm, informing a human operator.

Note that the first input state given to the agent is not required to be around the boundaries of the SOW, making it possible to update previously made predictions based on the current ground-truth state of the vessel trajectory to be imitated.

IV. Imitation Learning

A. Relationship with Reinforcement Learning

Imitation Learning is the general approach of extracting the underlying policy given a fixed dataset of expert trajectories. Those trajectories are sampled from an environment, which can be formalized as Markov Decision Process (MDP). This framework consists of an agent which observes the current state of the environment $s_t \in \mathcal{S}$ and interacts with it in discrete timesteps by choosing an appropriate action $a_t \in \mathcal{A}$ based on a policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$ depending on the learnable parameters θ. The environment then provides the next state s_{t+1} given the internal transition dynamics $p(s_{t+1} \mid s_t, a_t)$ and a scalar reward derived from a hand-crafted reward function $r(s_t, a_t)$ that indicates how "good" the proposed action was.

One way to solve such a sequential decision problem is Reinforcement Learning, which uses the feedback of the environment (the reward signal) to improve the policy with an effort to maximize the discounted sum of future rewards (the return). In fact, Reinforcement Learning has been used in the maritime domain for the path following problem [26] or continuous control [27]–[29]. However, these approaches either do not incorporate time (i.e. only the geometry of the curves is relevant, not the timing in tracking a timed refereced signal), or take place in the very distinct field of autonomous driving, where an agent learns optimal strategies to control a vessel in order to reach a destination under pre-defined premises such as taking the fastest route or consuming the least energy while avoiding restricted areas and collisions. These methods require a model of the physical interaction of the vessel with its environment.

Reinforcement Learning and Imitation Learning operate within the same mathematical framework of MDPs with the exception that in the case of Imitation Learning the reward function is not required. A key difference, as designing a suitable reward function by hand that implicitly defines the desired goal can be a tedious task, especially if the reward function is non-sparse (like the *Gaussian cross-track error* used in [26]) and the researcher tries to "guide" the agent (i.e. direct feedback at every timestep instead of just a single reward given at the very end of an episode based on the outcome).

B. Behavioral Cloning

Behavioral Cloning [30] reduces the task of imitating the behavior of an expert to a supervised learning problem, where a neural network π_θ parameterized by θ learns to map states to actions as closely as possible to the expert policy π_*. That is, the network learns to sample $\pi_\theta(s) \to a$ from the probability density $P_\theta(a|s)$ estimating the true density $P_*(a|s)$ of expert actions given the enviromental states. This estimation is done by maximum likelihood optimization

$$\text{argmin}_\theta \, \mathbb{E}_{(s,a_*)\sim P_*} - \log P_\theta(a_*|s), \tag{1}$$

where the latent features before the output layer of π_θ are passed as moments to a known probability density (such as a Gaussian with diagonal covariance) to construct a neural network for P_θ.

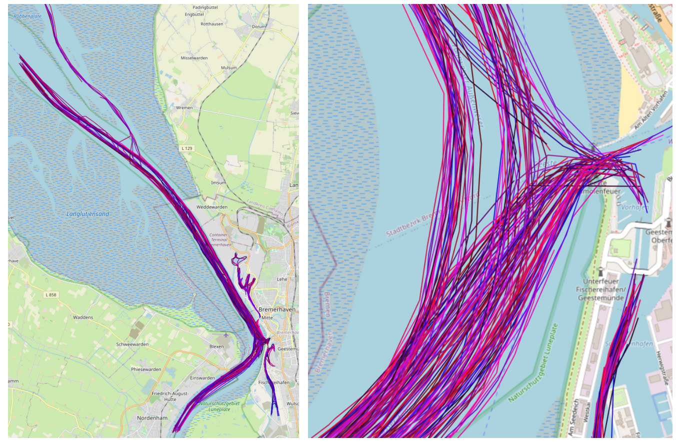
V. Experiments

A. Extracting Trajectories from AIS Data

We take historical AIS data from the coast of Bremerhaven from every first month of the quarters of the year 2020. The process of filtering and extracting single vessel trajectories is done in conjunction with the library *MovingPandas* [32].

First, AIS signals having speed-over-ground values too low ($<$ 3 knots; e.g. ships at moore) or unrealistically high ($>=$ 30 knots) are filtered out, for the ship types that we are most interested in, e.g., tankers, cargo or big cruise ships. Then, the records are aggregated and ordered in time by their respected Maritime Mobility Service Identity (MMSI), resulting in the raw trajectories. In order to smooth them out, the direction and speed information is interpolated. That is, we drop the original values for speed-over-ground and course-over-ground in favor of the interpolated $direction$ and $speed$, respectively.

To extract single voyages, the trajectories are then split by time gaps in consecutive AIS signals of more than 5 minutes, as well as by potential stops, anchoring or waiting in floodgates (ships that stay within the same area of 15 diameters for at least 3 minutes). We remove stops because the state representation does not include a specific time component, making consecutive stop states indistinguishable for the agent. Voyages with a length of less than 1.5 kilometers are removed to create a dataset that consists of expert demonstrations that span a rather long time horizon, in which underlying navigational rules are noticeable.



(a) (b)

Fig. 2. (a) Example of 200 extracted trajectories. Region of interest is within the bounds of longitude from $8°37.2'N$ to $8°58.8'N$ and latitude from $53°49'W$ to $53°72.2'W$. (b) zoom into the entrance to the double floodgate, displaying multiple motion clusters.

Afterwards, the records of each trajectory are resampled and linearly interpolated to have a fixed time interval of 10 seconds. This is done to implicitly incorporate a time scale into the system. The resulting dataset, after excluding ship types other than those defined by Appendix A, consists of 11,300 trajectories, with a subset being shown in Fig. 2(a).

B. Fusing non-kinematic information

Our raw AIS messages lack the static information, i.e. they do not contain any data about the ship type or size of the broadcasting vessel. A publicly available ship database service [33] is used to fuse additional information to the data records matching the given MMSIs. Consequently, the list of ship types included in the final dataset consist of the definitions given by this service.

In terms of weather conditions, the wind force and wind direction for the respective historical time frame is taken from the Climate Data Center of Germany's National Meteorological Service and the concrete weather station in Bremerhaven with ID 701. Tide information is taken from the European Commission Joint Research Centre and an openly accessible water level station in Cuxhaven [34].

Information about wind and tide gauge is fused to the nearest timestamp of the trajectory's data frame.

C. State Representation and Action Space

The state representation plays an important role in every Reinforcement Learning scenario. A corresponding feature space must include all information necessary for the agent to be able to choose a suitable next action without the need for past states. The agent's decisions made for the future purely depends on the state given in the present, with no history involved. As our dataset consists of an arbitrary amount of motion clusters, the state representation has to include, besides kinematic factors such as position, speed and direction, some information about the desired destination, namely the angle and distance, in order for the agent to distinguish and recognize its current motion cluster. Unfortunately, our initial dataset of raw AIS signals does not contain any information about the destination of the vessel. Moreover, the destination parameters of any AIS signal has to be set manually and is thus unreliable in the first place. For the purpose of this work, we artificially compute related features to destination based on the last position achieved per voyage.

Besides the features mentioned, additional information about present water level, wind force and wind direction are added to the state to capture potential underlying causalities in ship maneuvering during different environmental situations. At time step t, the state is then described as

$$\begin{aligned} s_t := \{&lon_t, lat_t, direction_t, speed_t, waterLevel_t,\\ &windForce_t, windDirection_t,\\ &angleToDestination_t, distanceToDestination_t\},\end{aligned}$$

with the vessel-related features representing the expert demonstrated behavior. On the other hand, the agent's actions are proposals of $course_t$ and $tempo_t$ (targetting $direction_t$ and $speed_t$, respectively) that are consistent with the next position $\hat{s}_{t+1}$ of the imitated vessel under a constant-velocity evolution rule between the two time steps of interest,

$$a_t := \{course_t, tempo_t\}.$$

That is, $\hat{s}_t$ represents the location of the imitated vessel as predicted by the agent, which is expected to be close to the corresponding information in the ground truth s_t.

D. Training Parameters

A fully connected neural network is used as internal regressor of Behavioral Cloning, whose architecture can be described by using the notation of $n_h^i = N$, where n is the i-th hidden layer with N neurons. The network of $n_h^1 = 512$, $n_h^2 = 256$, $n_h^3 = 128$ and $n_h^4 = 64$ is trained for 20 epochs with a learning rate of $\alpha = 10^{-7}$ and a batch size of 32. The Adam optimizer [35], a gradient-based algorithm to minimize the loss function, is used during training. The dataset is split into 9071 trajectories for training and 2267 trajectories for testing (where the prediction error is computed).

E. Evaluation Metric

To evaluate the performance of our agent, we follow the approach of [5] by calculating the average prediction error for each trajectory in the test set. Hereby, the *great-circle* distance d_H between the true and the predicted vessel positions on the earth's surface is derived from the *haversine formula* given the geographical coordinates of the two points s_1 and s_2:

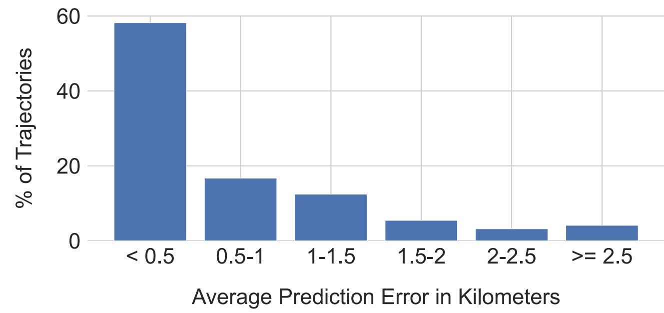


Fig. 3. Histogram showing the percentage of test trajectories and their corresponding average prediction error during validation.

$$d_H(s_1, s_2) = 2R \arcsin \sqrt{\sin^2 \tilde{\phi} + \cos\phi_1 \cos\phi_2 \sin^2 \cos\tilde{\lambda}}$$

where R is the radius of the earth, ϕ_1 and ϕ_2 are the latitude and λ_1 and λ_2 are the longitude values of points s_1 and s_2, $\tilde{\phi} = \frac{\phi_2 - \phi_1}{2}$, and $\tilde{\lambda} = \frac{\lambda_2 - \lambda_1}{2}$.

For every trajectory T in the test set, consisting of L_T total timesteps, the average prediction error based on the haversine distance, $APEH$, is then calculated as:

$$APEH_T = \frac{1}{L_T} \sum_{i=1}^{L_T} d_H(s_i, \hat{s}_i) \tag{2}$$

From this, the *overall* prediction error is defined to be the median of all APEHs of every trajectory in the test set.

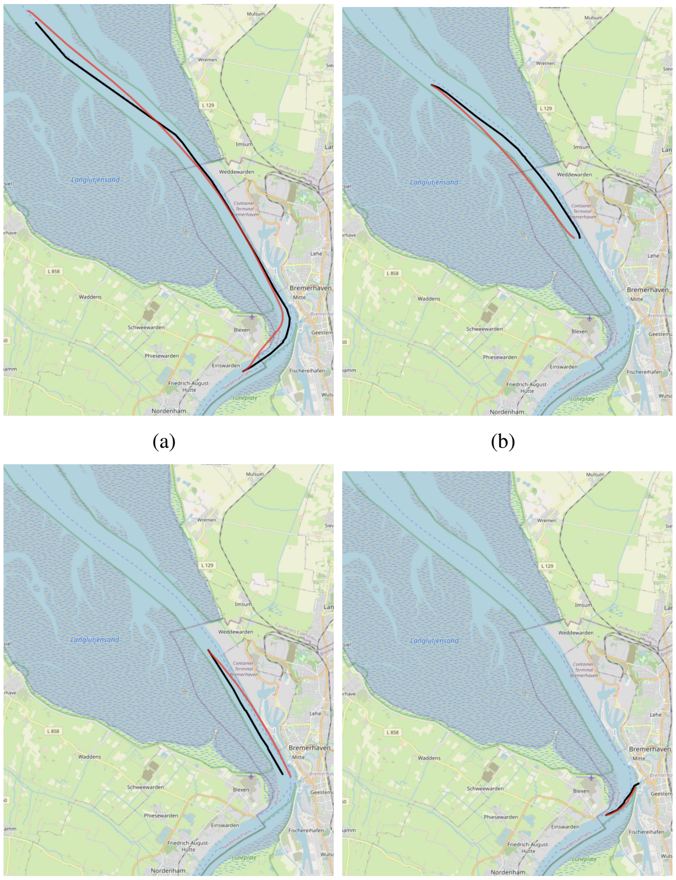
VI. Results and Discussion

We present a quantitative and qualitative analysis of the performance of our agent, mentioning possible improvements and opening the discussion about the desired accuracy of vessel predictions, as there is no consensus in the literature yet about the definition of what a "good" prognosis is.

The overall prediction error, which is the median of all APEHs, defined by Eq. 2, is 362 meters with a median absolute deviation of 250 meters. Figure 3 illustrates the percentage of all test trajectories within certain ranges of APEH. More than 58% of generated trajectories have an APEH of less than 500 meters, while 4% have an APEH higher than 2 kilometers.

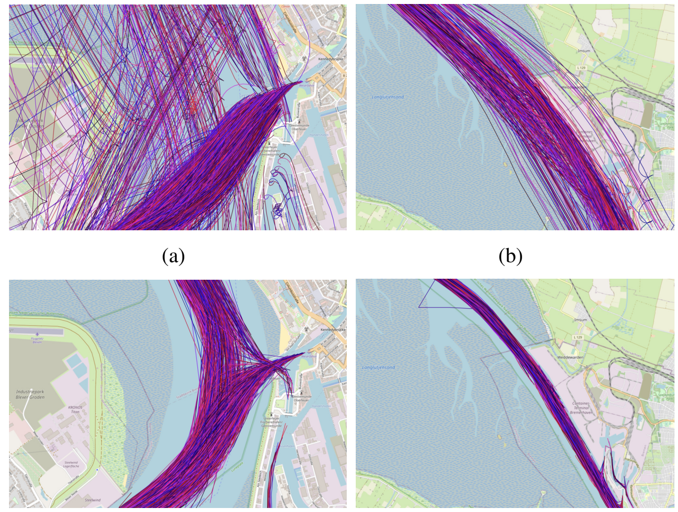
Setting a threshold for presumably "good" predictions is not straightforward, because multiple factors are involved, namely the pure ship size, size of the observation window, average gap size of passing ships, time horizon of the prediction and uncertainty of the AIS signals. In our scenario — the coast and port area of Bremerhaven — the observation window spans roughly 360 square kilometers. Internally, the agent is not aware of any restrictions, e.g., waterways. In fact, Figs. 5(a) and 5(b) reveal that many predicted trajectories go over land. Splitting the system observation window into multiple grid cells, that only span on water, will prevent unrealistic predictions and potentially improve the overall accuracy.

The average ship size of our dataset is 163 meters in length and 23 meters in width. It follows, that our overall



(a) (b) (c) (d)

Fig. 4. Supposedly "good" generated trajectories. Black lines indicate the ground-truth trajectories, while red lines indicate the agent's prediction. The APEHs and time horizons are 370 m and 68.5 min. for (a), 484 m and 31.16 min. for (b), 326 m and 22.8 min. for (c), and 141 m and 7.8 min for (d).



(a) (b) (c) (d)

Fig. 5. (a)-(b) 2000 generated trajectories by the agent (c)-(d) corresponding ground-truth trajectories of the test dataset .

prediction error is in the order of two average ship lengths. If the area of interest would be, for example, the Baltic Sea and the generated predictions would have time horizons of multiple hours, this could intuitively be labeled as an accurate system. However, in a close quartered area such as the port of Bremerhaven with small waterlines and ships passing one another with less than 50 meters distance, this accuracy is insufficient.

Nevertheless, the agent learns representative trajectories, meaning that it tries to extract knowledge and behavior of what an average human captain would do under certain conditions. As the train and test datasets contain anomalies and diverging behaviors from normality, the agent should not even replicate those occurrences, that in return cause worse prediction accuracy.

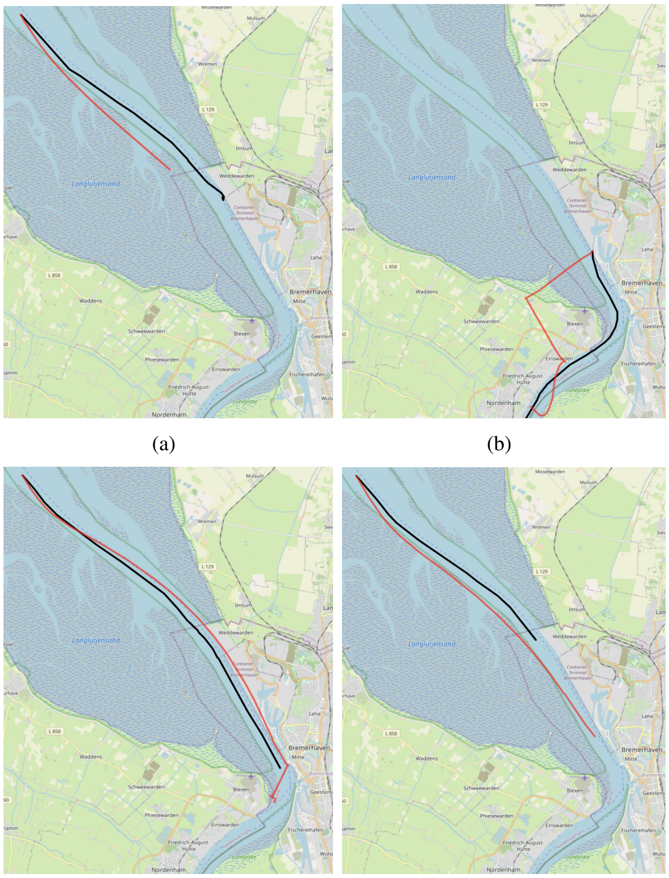
In order to get a better understanding whether the agent actually learned the data distribution and representative trajectories, we illustrate 2000 predicted trajectories and their respected ground-truth in Fig. 5. The Fig. 5(a) shows, that the agent reproduces the motion pattern of the two ferries. Though, the same panel also indicates that generated trajectories are all over the place, which differs significantly from the respective ground-truth data displayed in Fig. 5(c). Looking at the area further north in Figs. 5(b) and 5(d), we see a comparable motion cluster, that in the case of the agent predictions spreads a lot wider.

Single instances of predictions are displayed for good trajectories in Fig. 4 and supposedly bad trajectories with a high APEH in Fig. 6. Here, Fig. 4(a) is a very well predicted trajectory, starting in Nordenham and leaving Bremerhaven towards the North Sea. Figures 4(b) and 4(c) are instances where the APEH is low and therefore might be seen as accurate. This is however misleading since the average errors of these instances — although due to a sideways shift of the prediction, following the ground-truth almost in parallel — are 326 m and 141 m, respectively, way higher than the typical distances (< 50 m) at which ships pass close to each other, therefore deeming the predictions as inaccurate.

In contrast, Fig. 6 shows, except for 6(b), that predictions with some of the highest APEHs are not that far off the ground-truth, especially in the case of Fig. 6(c). Although the shapes of the predicted trajectories are similar, the agent did not adjust the speed (or more precise the *tempo*) correctly, which leads to situations where the agent is constantly trailing behind the ground-truth.

VII. Conclusion

We proposed an approach that utilizes Behavioral Cloning in conjunction with a 4-layer neural network as a method to predict end-to-end vessel trajectories based on single input states. Therefore, the system is more flexible than related work, dropping the need for past input sequences while at the same time allowing for variable output lengths. Results show that the agent is indeed capable of learning isolated motion patterns, e.g., the ferry ride between Bremerhaven and Nordenham or the exiting towards the North Sea. The overall prediction error of 362 meters is hard to categorize purely based on the raw



(a) (b) (c) (d)

Fig. 6. Supposedly "bad" generated trajectories. Black lines indicate the ground-truth trajectories, while red lines indicate the agent's prediction. The APEHs and time horizons are 2917 m and 29.1 min. for (a), 2315 m and 43.3 min. for (b), 2088 m and 70.8 min. for (c), and 2914 m and 51.3 min for (d).

number, as argued in the previous section. In its current form, the system is insufficient for maritime surveillance close to the port. However, this work is the first of its kind and by consecutive research and the investigation of improvements, the performance could increase to practical level to be usable in a real-world scenario.

A future research avenue contemplates using feature extractors more sophisticated than the simple multi-layer perceptrons used. We also invite researchers to follow up and propose a standard metric to evaluate vessel trajectory prediction systems.

References

[1] International Maritime Organization, "Revised guidelines for the onboard operational use of shipborne automatic identification systems (ais)," 2015.

[2] P. Last, C. Bahlke, M. Hering-Bertram, and L. Linsen, "Comprehensive analysis of automatic identification system (ais) data in regard to vessel movement prediction," *Journal of Navigation*, vol. 67, no. 5, p. 791–809, 2014.

[3] B. Kawan, H. Wang, G. Li, and K. Chhantyal, "Data-driven modeling of ship motion prediction based on support vector regression," 2017.

[4] International Maritime Organization, "Convention on the international regulations for preventing collisions at sea, 1972 (COLREGs)," 1972. [Online]. Available: https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx

[5] S. Capobianco, L. M. Millefiori, N. Forti, P. Braca, and P. Willett, "Deep learning methods for vessel trajectory prediction based on recurrent neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 4329–4346, 2021.

[6] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.

[7] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[8] H. Rong, A. Teixeira, and C. Guedes Soares, "Ship trajectory uncertainty prediction based on a gaussian process model," *Ocean Engineering*, vol. 182, pp. 499–511, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801818315427

[9] G. Pallotta, S. Horn, P. Braca, and K. Bryan, "Context-enhanced vessel prediction based on ornstein-uhlenbeck processes using historical ais traffic patterns: Real-world experimental results," in *17th international conference on information fusion (FUSION)*. IEEE, 2014, pp. 1–7.

[10] L. P. Perera, P. Oliveira, and C. G. Soares, "Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1188–1200, 2012.

[11] B. L. Young, "Predicting vessel trajectories from ais data using r," Naval Postgraduate School Monterey United States, Tech. Rep., 2017.

[12] C. Nishizaki, M. Terayama, T. Okazaki, and R. Shoji, "Development of navigation support system to predict new course of ship," in *2018 World Automation Congress (WAC)*. IEEE, 2018, pp. 1–5.

[13] Z. Xiao, X. Fu, L. Zhang, W. Zhang, R. W. Liu, Z. Liu, and R. S. M. Goh, "Big data driven vessel trajectory and navigating state prediction with adaptive learning, motion modeling and particle filtering techniques," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, "Prediction oof vessel trajectories from ais data via sequence-to-sequence recurrent neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8936–8940.

[16] R. W. Liu, M. Liang, J. Nie, Y. Yuan, Z. Xiong, H. Yu, and N. Guizani, "Stmgcn: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multi-graph convolutional network," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2022.

[17] J. Chen, X. Li, Y. Xiao, H. Chen, and Y. Zhao, "Fra-lstm: A vessel trajectory prediction method based on fusion of the forward and reverse sub-network," *arXiv preprint arXiv:2201.07606*, 2022.

[18] D. Alizadeh, A. A. Alesheikh, and M. Sharif, "Vessel trajectory prediction using historical automatic identification system data," *The Journal of Navigation*, vol. 74, no. 1, pp. 156–174, 2021.

[19] B. Zhang, Z. Xu, J. Zhang, and G. Wu, "A warning framework for avoiding vessel-bridge and vessel-vessel collisions based on generative adversarial and dual-task networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 5, pp. 629–649, 2022.

[20] S. Capobianco, N. Forti, L. M. Millefiori, P. Braca, and P. Willett, "Uncertainty-aware recurrent encoder-decoder networks for vessel trajectory prediction," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. IEEE, 2021, pp. 1–5.

[21] ——, "Recurrent encoder-decoder networks for vessel trajectory prediction with uncertainty estimation," *arXiv preprint arXiv:2205.05404*, 2022.

[22] S. Zhang, L. Wang, M. Zhu, S. Chen, H. Zhang, and Z. Zeng, "A bidirectional lstm ship trajectory prediction method based on attention mechanism," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5. IEEE, 2021, pp. 1987–1993.

[23] D.-D. Nguyen, C. Le Van, and M. I. Ali, "Vessel destination and arrival time prediction with sequence-to-sequence models over spatial grid," in *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*, 2018, pp. 217–220.

[24] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International Journal of Forecasting*, vol. 36, 07 2019.

[25] H.-Y. S. Chien, J. Turek, N. M. Beckage, V. A. Vo, C. J. Honey, and T. L. Willke, "Slower is better: Revisiting the forgetting mechanism in lstm for slower information decay," *ArXiv*, vol. abs/2105.05944, 2021.
[26] A. B. Martinsen and A. M. Lekkas, "Curved path following with deep reinforcement learning: Results from three vessel models," in *OCEANS 2018 MTS/IEEE Charleston*. IEEE, 2018, pp. 1–8.
[27] M. Etemad, N. Zare, M. Sarvmaili, A. Soares, B. B. Machado, and S. Matwin, "Using deep reinforcement learning methods for autonomous vessels in 2d environments," *arXiv preprint arXiv:2003.10249*, 2020.
[28] N. Zare, B. Brandoli, M. Sarvmaili, A. Soares, and S. Matwin, "Continuous control with deep reinforcement learning for autonomous vessels," *arXiv preprint arXiv:2106.14130*, 2021.
[29] S. Guo, X. Zhang, Y. Zheng, and Y. Du, "An autonomous path planning model for unmanned ships based on deep reinforcement learning," *Sensors*, vol. 20, no. 2, 2020.
[30] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1. Morgan-Kaufmann, 1988.
[31] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
[32] A. Graser, "Movingpandas: efficient structures for movement data in python," *GIForum*, vol. 1, pp. 54–68, 2019.
[33] MyShipTracking. Vessel database. [Online]. Available: https://www.myshiptracking.com/vessel
[34] European Commission Joint Research Centre - ISPRA - Space, Security and Migration Directorate. Tide gauge details - germany - cuxhaven. [Online]. Available: https://webcritech.jrc.ec.europa.eu/SeaLevelsDb/Device/853
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

APPENDIX

Custom definitions of ship types based on the used vessel database [33] present in the dataset:

'Bulk Carrier', 'Bunkering Tanker', 'Cargo', 'Cargo A', 'Cargo B', 'Cargo C', 'Cargo D', 'Cement Carrier', 'Chemical Tanker', 'Container Ship', 'Edible Oil Tanker', 'General Cargo', 'Heavy Lift Vessel', 'Heavy Load Carrier','Lpg Tanker', 'Oil Products Tanker', 'Oil/Chemical Tanker', 'Passenger', 'Passengers Ship', 'Ro-Ro Cargo', 'Ro-Ro/Passenger Ship', 'Suction Dredger', 'Tanker', 'Tanker A', 'Tanker B', 'Trailing Suction Hopper Dredger','Vehicles Carrier', 'Waste Disposal Vessel'