

Short-term Inland Vessel Trajectory Prediction with Encoder-Decoder Models

Kathrin Donandt¹, Karim Böttger², Dirk Söffker³

Abstract—Accurate vessel trajectory prediction is necessary for safe and efficient navigation. Deep learning-based prediction models, esp. encoder-decoders, are rarely applied to inland navigation specifically. Approaches from the maritime domain cannot directly be transferred to river navigation due to specific driving behavior influencing factors. Different encoder-decoder architectures, including a transformer encoder-decoder, are compared herein for predicting the next positions of inland vessels, given not only spatio-temporal information from AIS, but also river specific features. The results show that the reformulation of the regression task as classification problem and the inclusion of river specific features yield the lowest displacement errors. The standard LSTM encoder-decoder outperforms the transformer encoder-decoder for the data considered, but is computationally more expensive. In this study for the first time a transformer-based encoder-decoder model is applied to the problem of predicting the ship trajectory. Here, a feature vector using the river-specific context of navigation input parameters is established. Future studies can build on the proposed models, investigate the improvement of the computationally more efficient transformer, e.g. through further hyper-parameter optimization, and use additional river-specific information in the context representation to further increase prediction accuracy.

Index Terms—vessel trajectory prediction, encoder-decoder model, transformer, inland navigation

I. INTRODUCTION

Predicting the driving behavior of vessels accurately is crucial to avoid collisions and find an efficient route when navigating in confined spaces. Automatic prediction can support navigators whereas autonomous vessels are completely dependent on it to determine the path to follow. Ship models, which can predict a vessel’s trajectory with high precision by considering the individual ship’s driving dynamics in a given waterway condition, are expensive to be obtained and are mostly used in simulators. In real world applications that include trajectory prediction, simple physical models, such as constant velocity models (CVM), still prevail [1], [2]. Kalman filters have been suggested to overcome the linearity constraint of CVMs but have limited prediction horizons [1]. With the rising AIS data availability in the last years, data-driven models, including deep neural networks, have gained

popularity. They have been studied as an alternative or in addition to model-based approaches to deal with complex navigation situations and longer prediction horizons. Deep learning (DL)-based trajectory prediction has mainly been applied to ocean-going vessels using spatio-temporal vessel data alone. This is problematic in the inland shipping domain, where waterway characteristics such as river flow and river shape restrict the vessels’ navigation. In this study, such contextual information is taken into consideration by adapting the common initial decoder input of encoder-decoder models. Encoder-decoder models have in general shown superior performance for sequence-to-sequence tasks compared to classical recurrent neural networks (RNNs). The advantage of these models over RNNs which require a recursive strategy to predict several time steps into the future has already been confirmed for the inland domain as well (see [3]). Therefore, encoder-decoder models are used herein to predict a sequence of next positions of a vessel’s trajectory given a sequence of past positions. Transformer encoder-decoder models, that have not yet been applied to trajectory prediction in navigation¹, are investigated.

A brief overview of related works in the field of DL-based trajectory prediction for vessels specifically, and transformer-based approaches for trajectory prediction in general, is given in the following section. In Section III, encoder-decoders are shortly explained before the inland-specific, context-sensitive classification transformer is introduced in Section IV. Section V is dedicated to the experiment details, evaluation methods and results, before a conclusion and outlook is given in Section VI.

II. RELATED WORK

A. Deep learning-based vessel trajectory prediction

Deep learning has been successfully applied for trajectory prediction of humans, vehicles and maritime vessels [5]–[10]. Inland navigation-specific DL approaches, are, however, rare. Encoder-decoder architectures have gained popularity in the last years in the navigation domain [2], [3], [11]–[14]. For inland navigation specifically, only two studies [3], [14] exist in which this neural network architecture is suggested. In [3], different RNN encoder-decoder models without attention process spatio-temporal data of an inland vessel of the last 10 min to predict its positions for the upcoming 5 min. The proposed model outperforms the RNN baseline which predicts the output sequence by repeatedly

¹K. Donandt is with the Institute of Ship Technology, Ocean Engineering and Transport Systems (ISMT), University of Duisburg-Essen, Duisburg, Germany. kathrin.donandt@uni-due.de

²K. Böttger is with the German Federal Waterway Engineering and Research Institute. karim.boettger@baw.de

³D. Söffker is with the Chair of Dynamics and Control (SRS), University of Duisburg-Essen, Duisburg, Germany. soeffker@uni-due.de

*This publication originates from a joint research project between the German Federal Waterways Engineering and Research Institute and the ISMT, and was written in collaboration with the SRS.

¹The study of [4], see Sec. II. for details, has not yet been published.

feeding the last output back to the model as new input sequence element.

Waterway specific information in addition to spatio-temporal vessel data has rarely been considered in the literature on DL-based vessel trajectory prediction. In [14], radar images are mapped to electronic navigational charts (ENC) by a convolutional neural network in order to generate a feature vector representing the navigation context. This feature vector is then fed to an LSTM encoder-decoder for inland vessel trajectory prediction.

No homogeneous understanding of “short-term” and “long-term” prediction in inland navigation exist in the literature covering DL-based approaches. For anticipation of future ship locations in the context of autonomous navigation, [14] use input and output sequences comprising 15 s and 150 s, respectively, and a time interval size between consecutive time steps of 3 s. The authors speak of “long-term” prediction probably due to the high number of prediction steps with the given time step size. In [3], short-term prediction horizons are defined to range from 5 to 15 min in the context of collision risk alert, the time interval size is set to 30 s and an input sequence of 20 data points (10 min) is fed to the predictor. Finally, a trajectory segment of up to 10 min is reconstructed given an input sequence of 4 min before or after this segment, using a time interval size of 1 min [15].

In the present work, the observation and prediction windows are defined similar to [3] and [15]. Different input and output sequence lengths combinations are tested. During training, either 5 or 10 min are fed to the prediction model to obtain the next 15 or 30 min. Thus, the effect of varying the input length as well as the output length can be analyzed. The unusually long prediction horizon of 30 min is only used for model training, and the errors in a prediction window of up to 15 min are compared to the errors of models trained with smaller prediction windows.

B. Transformer for trajectory prediction

Transformer encoder-decoder models based on attention mechanisms [16] have shown superior performance when compared to the standard long short-term memory (LSTM) encoder-decoder models in the Natural Language Processing field, where it has first been applied. It has already been employed to trajectory prediction for humans and automotive vehicles [17]–[19]. For vessel trajectory prediction, a transformer that predicts a single time step ahead is presented in [4]. A transformer encoder-decoder model mapping an input to an output time series has not been investigated yet.

III. METHODOLOGICAL BACKGROUND

Encoder-decoder models have originally been developed for machine translation [20], but are nowadays applied to other sequence-to-sequence modelling problems as well. The architecture of a classical encoder-decoder model is depicted in Fig. 1. The information of the whole input sequence x_1, \dots, x_n is encoded by an RNN into a so-called context-vector which is fed to the decoder as first hidden state h_0^d . In the classical model, the context vector simply consists of the

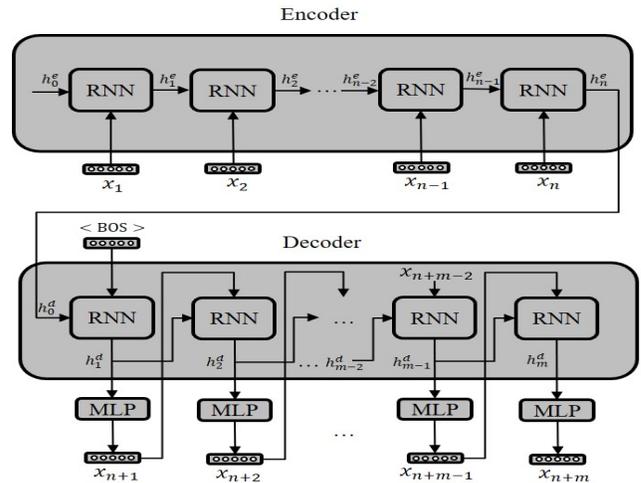


Fig. 1: Classical encoder-decoder architecture

last encoder hidden state such that $h_n^e = h_0^d$. The decoder generates the output sequence x_{n+1}, \dots, x_{n+m} by another RNN, receiving as input at time step t the previous decoder output. The first decoder input commonly consists in a placeholder (commonly referred to as <BOS> (“beginning-of-sequence”)) token). The hidden state obtained after processing one time step in the decoder-RNN is fed to the output layer (e.g. a multilayer perceptron (MLP)) to obtain the prediction. More advanced encoder-decoders use aggregation functions that process all hidden states of the encoder to generate the context vector. This is expected to improve the usage of information from the encoded sequence in the decoder. Attention-based aggregation functions are commonly used. They are able to consider the individual significance of each input when generating the output sequence. Transformers [16] are state of the art encoder-decoders substituting the encoder- and decoder-RNNs by stacked attention and feedforward layers. The sequentiality of the input and output sequences is taken into consideration by positional encodings.

IV. CONTEXT-SENSITIVE CLASSIFICATION TRANSFORMER

The overall architecture of the transformer encoder-decoder model for inland vessel trajectory prediction developed in this study is shown in Fig. 2. It differs from the general transformer architecture only in the input and output sequence processing layers preceding the actual transformer-specific model parts (positional encoding, transformer encoder and decoder). For details on the inner structure of these transformer-specific components, the interested reader is referred to [16]. The proposed model receives a time series of $n \in \{5, 10\}$ travelled distances and course over ground (COG) changes as input, and generates the next $m \in \{15, 30\}$ distances and COG changes. The time interval size is 1 min, and three combinations of n and m are considered (5-15, 10-15 and 10-30). A sequence fed to the encoder-decoder model is defined as

$$x_1, \dots, x_{n+m},$$

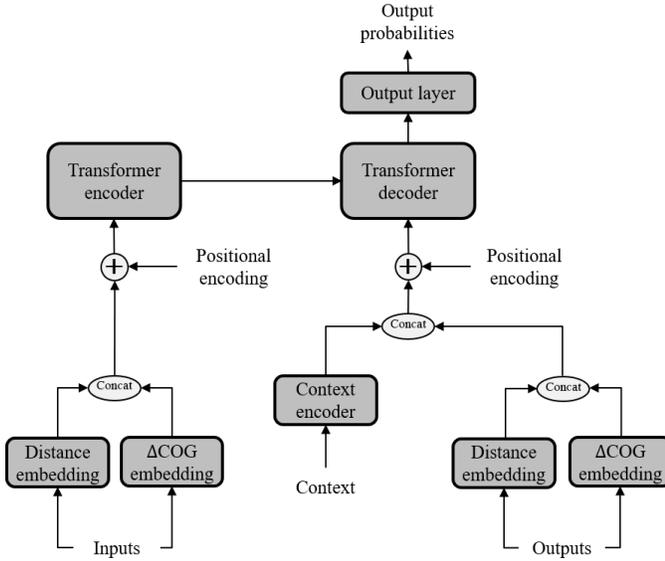


Fig. 2: Context-sensitive classification transformer

where

$$x_t = (|pos_{t+1} - pos_t|, cogDiff(t))$$

$$\forall t \in \{1, \dots, n + m\}$$

and

$$cogDiff(t) \begin{cases} +w, & \text{clock-wise change} \\ -w, & \text{counter-clock-wise change} \end{cases}$$

with

$$w = \min\{|cog_{t+1} - cog_t|, 360 - |cog_{t+1} - cog_t|\}.$$

For simplicity, 180° is assumed to be the maximum change of the course over ground (COG) in two consecutive time steps. The distance and ΔCOG embedding layers encode $|pos_{t+1} - pos_t|$ and $cogDiff(t)$, respectively, into high-dimensional feature vectors of these input and output components.

The original regression problem of predicting the next position is reformulated as classification problem, thus the output sequence is obtained from output probabilities of classes and not directly. Due to this reformulation, an adequate loss function for classification models needs to be employed. The overall loss of the proposed classification model is therefore defined as

$$\mathcal{L}_{total} = \frac{1}{2}CEL_{cogDiff} + \frac{1}{2}CEL_{distance},$$

where $CEL_{cogDiff}$ and $CEL_{distance}$ calculate the cross entropy loss (CEL) of the predicted probability distributions for the COG changes and the distances, respectively.

The classification reformulation is inspired by the works of Nguyen et al. [4], [11]. By obtaining a probability distribution as output, it is possible to sample several possible output sequences and determine the model's uncertainty. The encoder-decoder model could thus be part of hybrid model that falls back to an alternative prediction model in case of

high uncertainty. In this study, a greedy decoding method is used for simplicity instead of sampling. It picks the most likely output class at each predicted time step to generate a single output sequence per input.

To obtain the class labels, the continuous distance and COG change values are discretized with a resolution of 1 m or 10 m (depending on the prediction horizon) and 1° , respectively. The euclidean distance between the last position pos_{n+m+1} of the original (interpolated) AIS data and the last position pos'_{n+m+1} , which is obtained by calculating the coordinates of the sequence using the initial position pos_1 , and the $n + m$ discretized distance and COG change values, is in average 25 m after 20 min and 70 m after 40 min. This is still less than the length of most commercial inland vessels. The deviation of the trajectory obtained with discretized distances and COG changes from the trajectory of the AIS data can be seen in Fig. 3 (blue and green lines).

The $\langle\text{BOS}\rangle$ token commonly used as first input to the decoder is replaced by the last encoder input in [2], where a bi-directional LSTM encoder-decoder is proposed for vessel trajectory prediction. In the present study, a vector representation of river-specific information is taken instead. It is expected that the inclusion of such context information improves the prediction performance, as inland navigation is constrained by several and frequently changing waterway conditions. Whereas only river radii are considered in this study, the approach is not limited to this specific river features, but can be extended to also consider water levels, flow velocities, fairway widths etc. For a specific sequence x_1, \dots, x_{n+m} , the context information vector is defined as

$$(cv(hm(pos_1) + 0.1 \times 0), cv(hm(pos_1) + 0.1 \times 1), \dots, cv(hm(pos_1) + 0.1 \times N)),$$

with

$$N = (7.7 \times (m + n) \times 60)/100,$$

where cv is a function returning the river curvature value given a specific hectometer, and hm is the hectometer function returning the hectometer for a given position. Thus, the context information is a sequence of cv values for each 100 m starting from the hectometer of the sequence's first position $hm(pos_1)$ and ending at the hectometer that can be reached within $n+m$ min with a maximum speed of 15 knots (7.7 m/s). For simplicity, it is assumed that the difference between this hectometer range and the real travelled distance is neglectable in the context of river-specific information representation. The context encoder consists of an embedding and feedforward layer. The curvature function values are discretized and passed through the embedding layer. The resulting N embeddings are concatenated and passed through the feedforward layer to obtain the final context vector representation used as first decoder input.

V. EXPERIMENTS AND RESULTS

A. Data details and pre-processing

The data set used in this study was provided by the German Federal Waterway Engineering and Research Institute

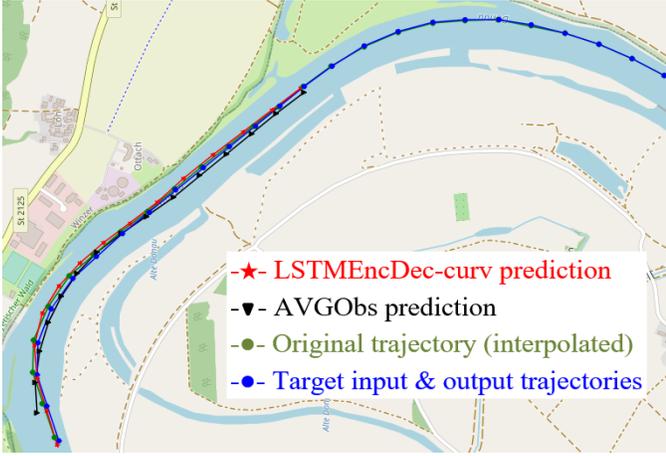


Fig. 3: Prediction example for $n = 10$ and $m = 15$.

(BAW) and comprises 29 months (05/2019 - 09/2021) of AIS data of a free-flowing section of the Danube river (hectometer range 2231.4 to 2321.44). A number of 7.7k trajectories with increasing hectometer direction are extracted (no turning allowed). As the vessel driving behaviors in free-flowing rivers differ significantly depending on the trajectory direction in relation to the river flow (up- or downhill), direction-specific models are developed in this study.

The time steps are standardized through linear interpolation, resulting in constant velocity and course over ground between consecutive AIS data points. The error caused by this interpolation is assumed to be neglectable for time intervals of up to 120 s. No interpolation is applied between consecutive data points with a larger time interval. From the interpolated trajectories, between 250k and 280k sequences of 20, 25, and 40 data points are extracted using a sliding window. Trajectory parts for which no interpolation is possible are not considered. The obtained sequences are split into training (80%), validation (10%) and test set (10%). Sequences containing time intervals of more than 2 min (i.e. trajectory parts that could not be interpolated) are not allowed.

For $post_t$, the AIS latitude and longitude values are converted to planar coordinates to enable euclidean distance calculations. The cog_t values are calculated given the planar coordinates and not taken from the interpolated AIS data directly to avoid inconsistencies.

The radii for each hectometer necessary to generate the context vector (see Sec. IV) were also provided by BAW. The river curvature function cv is obtained by inverting the radii values (to avoid infinity values) and interpolation. The hectometer function hm uses the hectometers provided by BAW for each AIS data point.

B. Model variants

The classification transformer model described in Sec. IV is compared to a combination of a regression and a classification model. The loss of this combined model is

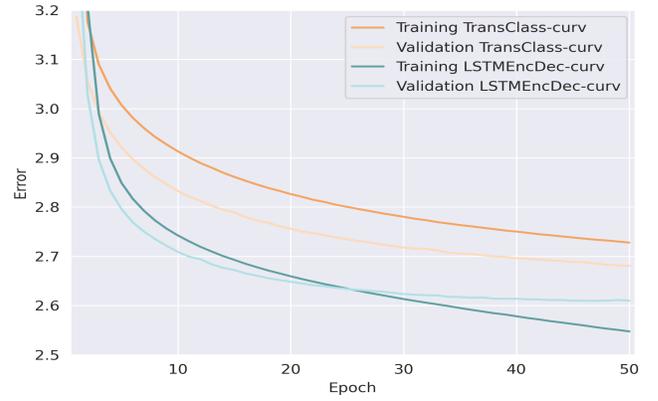


Fig. 4: Training and validation loss curves.

defined as

$$\mathcal{L}_{total} = \alpha \left(\frac{1}{2}CEL_{cogDiff} + \frac{1}{2}CEL_{distance} \right) + (1 - \alpha) \left(\frac{1}{2}MSE_{cogDiff} + \frac{1}{2}MSE_{distance} \right),$$

where $CEL_{cogDiff}$ and $CEL_{distance}$ are the cross entropy losses of the classification sub-model's predictions, $MSE_{cogDiff}$ and $MSE_{distance}$ the mean square errors of the regression sub-model's predictions, and α is a weighting factor and set to 0.5 in this study. The motivation of adding a regression sub-module is to give the classification model a notion of the numerical relation between the class labels. Typical classification problems do not assume a numerical relation between class labels.

To determine the performance gain when using a vector representation of the river curvature values as initial decoder input, the context-agnostic counterpart of the suggested model (i.e. using the default <BOS> embedding as initial decoder input) is also tested.

C. Comparison to LSTM encoder-decoder

A LSTM encoder-decoder without any aggregation function is compared to the suggested transformer model. The input and output sequences as well as the context information are processed by the corresponding embedding layers and context encoder as in the transformer (see Fig. 2) before being fed to the encoder and decoder. The loss function is identical to the one proposed for the transformer, and a context-agnostic LSTM encoder-decoder is tested as well.

D. Model training

For both the context-sensitive LSTM and transformer encoder-decoder (LSTMEncDec-curve and TransClass-curve) approaches, the hyper-parameters have to be defined. The best performance is obtained with the following configurations:

- Dropout: 0.1
- Hidden layer size (i.e. the number of hidden state features of the LSTM and the size of the fully connected layer in the transformer): 512

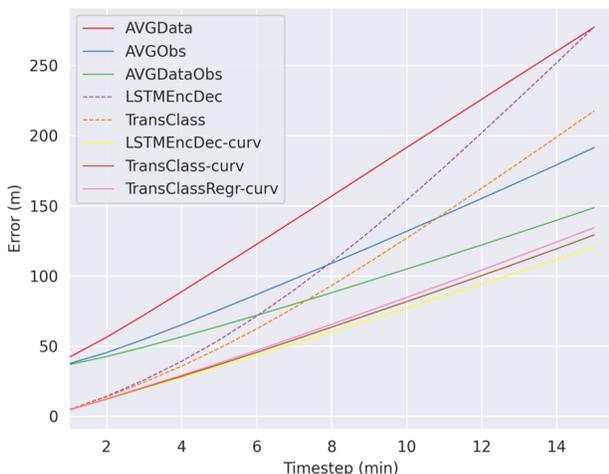


Fig. 5: Mean euclidean distance from target position.

TABLE I: Error (in meters) after 15 minutes.

Model	Mean	σ	Median
AVGDataObs	148.57	175.51	95.60
LSTMEncDec-curv	120.61	171.09	74.10
TransClass-curv	129.14	173.89	83.12

- Number of encoder/decoder layers: 3/3
- Size of distance/ Δ COG embedding (cf. Fig. 2): 512/512 (LSTMEncDec-curv) and 256/256 (TransClass-curv)
- Optimizer: Adam

The LSTMEncDec-curv model reaches a lower validation error and learns faster in terms of required number of epochs as shown in Fig. 4. The duration of one training epoch parallelized on two graphical cards are 7-17 min for the LSTMEncDec (depending on n and m) and around 2 min for the transformer model.

E. Comparison to average velocity models

Three average velocity models are used as baselines. The next positions are calculated by iteratively adding a specific distance d_i to the hectometer of the current position pos_i ($i \in n+1, \dots, n+m$) and returning the coordinates of the resulting point on the river axis. These baseline thus just give a rough estimate of where the vessel is located in terms of how far it gets during the prediction horizon. Note, that for a more sophisticated baseline, the offset of the river axis should be considered as well. The three baseline are:

- 1) Average observed velocity (AVGObs) - d_i is the mean distance travelled during observation and thus constant.
- 2) Average hectometer-specific velocity (AVGData) - d_i is the average distance travelled per minute at the current hectometer $hm(pos_i)$, obtained from all vessels passing this hectometer in the training data set.
- 3) Average hectometer-specific velocity with current deviation (AVGDataObs) - d_i is the average hectometer-specific velocity plus the mean deviation of the hectometer-specific velocities during observation.

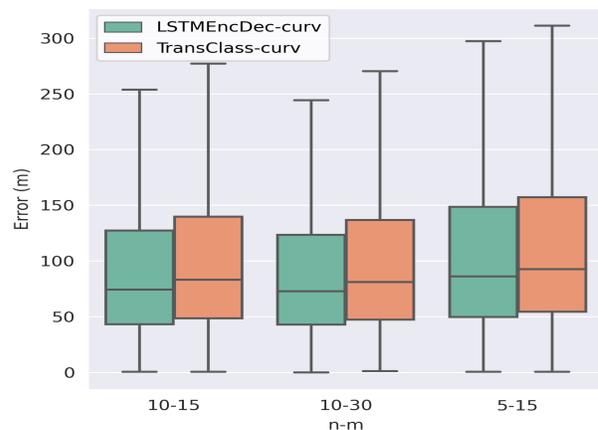


Fig. 6: Distribution of euclidean distances between predicted and target position after 15 min of models trained with varying observation (n) and prediction sequence lengths (m).

F. Prediction results

The prediction error of the transformer model, its variants, the LSTM encoder-decoder and the baselines is calculated as mean euclidean distance between the predicted and target position per time step and depicted in Fig. 5 for the case of $n = 10$ and $m = 15$. Note that the target positions are $pos'_t \forall t \in \{n+2, \dots, n+m+1\}$ and not the (interpolated) AIS data positions pos_t (see Sec. IV). The lowest errors are obtained by the context-sensitive models. TransClass-curv is outperformed by LSTMEncDec-curv. After 15 min, the error of LSTMEncDec-curv is 120.61 m (Table I). The addition of a regression sub-module (TransClassRegr-curv) does not improve the prediction. The errors of the proposed models and the baselines grow with a nearly constant factor when increasing the prediction horizon, whereas the context-agnostic transformer and LSTM encoder-decoders (TransClass and LSTMEncDec) show an exponential error development and perform worse than the baselines for a prediction horizon of more than 7 and 6 min, respectively. The improvement obtained by the inclusion of context information is higher for LSTMEncDec-curv.

A comparison of the context-sensitive classification models' performance with respect to varying observation and prediction horizons (n and m) is given in Fig. 6. Decreasing the observation horizon results in higher prediction errors. Training with longer prediction horizons slightly improves the models' performance for the given prediction horizon (15 min). LSTMEncDec-curv outperforms TransClass-curv for all settings.

An example of a trajectory obtained by LSTMEncDec-curv is visualized in Fig. 3. The deviation of the blue trajectory (target data trajectory $pos_1, pos'_2, pos'_3, \dots, pos'_{n+m+1}$) from the green one (underlying trajectory from AIS obtained after interpolation $pos_1, pos_2, \dots, pos_{n+m+1}$) is due to the discretization of the distances and COG changes as explained in Sec. IV.

Comparison of research results in vessel trajectory prediction is still challenging due to missing benchmarks and

TABLE II: Comparison to multi-task encoder decoder [14].

	Multi-task encoder-decoder	Context-sensitive encoder-decoders	
Model type	LSTM encoder-decoder + convolutional module	LSTM encoder-decoder	Transformer encoder-decoder
Covered region	Germany, Belgium, Netherlands	Danube km 2245-2321.2, uphill	
Data source	AIS, radar images, ENC	AIS	
Dataset size	44 h trajectories	272,573 sub-trajectories ($n = 10, m = 15$)	
Time step size	3 s	1 min	
Input/output sequence length	5/50 (15 s/2.5 min)	10/15 (10 min/15 min)	
ATE ² after 150 s	17.13 m	11.71 m	11.88 m

the usage of custom datasets, input/output sequence lengths, temporal resolution, and error measurements throughout the literature. Compared to the the multi-task LSTM encoder-decoder proposed in [14], LSTMEncDec-curve and TransClass-curve obtain a lower ATE² after 2.5 min in spite of the lower temporal resolution (see Table II for details). One reason might be the higher input/output sequence length ratio, another that the models herein are optimized for of a specific river section and navigation direction.

VI. CONCLUSION

Different encoder-decoder models are used for predicting the time series of the next 15 min of a vessel's trajectory after having observed 5 to 10 min. The LSTM encoder-decoder outperforms the transformer model, but is computationally more expensive. Using a context vector as first decoder input that represents the radii of the river section the vessel is heading the performance is significantly improved. The classification reformulation of the regression problem is necessary to successfully train the transformer model. The error introduced by the discretization of distance and COG change is assumed to be neglectable for the prediction horizon under consideration, but would be unacceptable in applications with high precision requirements, e.g. collision avoidance maneuver prediction. It should, however, be ensured that the positions obtained after discretization are located in the navigable area and not outside, which is theoretically what is possible with the current implementation. The proposed models outperform the baselines and the multi-task encoder-decoder [14]. In future works, the effect of enriching the context vector with further waterway-specific information needs to be investigated. Moreover, substituting the greedy decoding method used to generate the output sequence from the output probabilities by a beam search decoding could possibly lead to better results. Sampling output sequences would be required to get an estimate of the model's uncertainty. Finally, due to the sensitivity of deep neural networks to hyper-parameters, further hyper-parameter optimization needs to be conducted.

REFERENCES

[1] B. Murray and L. P. Perera, "An AIS-based deep learning framework for regional ship behavior prediction," *Reliability Engineering & System Safety*, vol. 124, no. 5, p. 107819, 2021.

²Absolute trajectory errors, given by the root of the average of the squared euclidean distances between prediction and target position at every time step (error measure used in [14]).

[2] 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.

[3] L. You, S. Xiao, Q. Peng, C. Claramunt, X. Han, Z. Guan, and J. Zhang, "ST-Seq2Seq: A Spatio-Temporal Feature-Optimized Seq2Seq Model for Short-Term Vessel Trajectory Prediction," *IEEE Access*, vol. 8, pp. 218 565–218 574, 2020.

[4] D. Nguyen and R. Fablet, "TrAISformer-a generative transformer for ais trajectory prediction," *ArXiv*, vol. abs/2109.03958, 2021.

[5] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[6] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakis, "Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2022.

[7] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, "Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1559–1582, 2018.

[8] S. Azimi, J. Salokannel, S. Lafond, J. Lilius, M. Salokorpi, and I. Porres, "A survey of machine learning approaches for surface maritime navigation," in *Maritime Transport VIII: Proceedings of the 8th International Conference on Maritime Transport (Maritime Transport '20)*, Barcelona, 2020, pp. 103–117.

[9] Z. Xiao, X. Fu, L. Zhang, and R. S. M. Goh, "Traffic Pattern Mining and Forecasting Technologies in Maritime Traffic Service Networks: A Comprehensive Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1796–1825, 2020.

[10] J. Jin, W. Zhou, and B. Jiang, "An Overview: Maritime Spatial-Temporal Trajectory Mining," *Journal of Physics: Conference Series*, vol. 1757, no. 1, p. 012125, 2021.

[11] D.-D. Nguyen, C. Le Van, and M. I. Ali, "Vessel Trajectory Prediction using Sequence-to-Sequence Models over Spatial Grid," in *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems (DEBS '18)*, New York, 2018, pp. 258–261.

[12] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, "Prediction of Vessel Trajectories From AIS Data Via Sequence-To-Sequence Recurrent Neural Networks," in *2020 IEEE Int. Conference on Acoustics, Speech and Signal Processing*, Barcelona, 2020, pp. 8936–8940.

[13] J. Sekhon and C. Fleming, "A Spatially and Temporally Attentive Joint Trajectory Prediction Framework for Modeling Vessel Intent," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, PMLR, 2020, vol. 120, pp. 318–327.

[14] P. Dijt and P. Mettes, "Trajectory Prediction Network for Future Anticipation of Ships," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, Dublin, 2020, pp. 73–81.

[15] Z. Yuan, J. Liu, Y. Liu, Q. Zhang, and R. W. Liu, "A multi-task analysis and modelling paradigm using LSTM for multi-source monitoring data of inland vessels," *Ocean Engineering*, vol. 213, p. 107604, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 30, Long Beach, USA, 2017, pp. 6000–6010.

[17] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[18] W. Chen, F. Wang, and H. Sun, "S2TNet: Spatio-Temporal Transformer Networks for Trajectory Prediction in Autonomous Driving," in *Proceedings of the 13th Asian Conference on Machine Learning*, vol. 157, 2021, pp. 454–469.

[19] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction," in *Computer Vision – ECCV 2020, 16th European Conference, Proceedings, Part XII*, Glasgow, UK, 2020, pp. 507–523.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Proceedings of the 27th Int. Conference on Neural Information Processing Systems*, vol. 2, Cambridge, USA, 2014, p. 3104–3112.