Sensors and Materials, Vol. 35, No. 9 (2023) 3393–3403
MYU Tokyo

S & M 3399

Enhancing Vessel Trajectory Prediction via Novel Loss Function in Deep Learning Model

Seung Bae Jeon,[1] Myeong-Hun Jeong,[1*†] Tae-young Lee,[1] and Dooyong Cho[2**†]

[1]Department of Civil Engineering, Chosun University,
309 Pilmun-daero, Dong-gu, Gwangju 61452, Republic of Korea
[2]Department of Convergence System Engineering, Chungnam National University,
99 Daehak-ro, Yuseong-gu, Daejeon 34134, Republic of Korea

(Received April 24, 2023; accepted August 18, 2023)

Keywords: loss function, deep learning, vessel trajectory prediction, automatic identification system data

Recent developments in data collection technology and sensor precision have led to the generation of large amounts of high-quality data. The vast vessel trajectory data obtained from precise automatic identification system data facilitate the development of marine-related research fields. In particular, vessel trajectory prediction, such as preventing risks in advance or providing efficient routes by predicting the vessel location, is one of the essential parts of advanced vessel traffic service. In this study, the vessel trajectory was accurately and robustly predicted using a novel loss function. In previous studies, the loss function was designed to minimize the distance between the destination and predicted location of vessels, whereas the proposed loss function was designed to minimize the area of the triangle formed by the origin, destination, and predicted location. In experiments, the proposed approach outperformed the state-of-the-art method, reducing the mean absolute error by 12%.

1. Introduction

Technologies for the efficient and safe management of marine resources and shipping networks have long attracted attention worldwide. In particular, various vessel traffic services (VTSs) that provide navigational traffic information, such as ship operation status or vessel traffic control to prevent maritime accidents from departure from the route, approach to a dangerous area, and risk of collision, have been proposed with the rapid growth of high-quality data owing to improvements in the data collection technology and sensor accuracy in the marine field. For instance, maintaining an efficient vessel traffic network through advanced VTSs can significantly contribute to economic development. In addition, it can provide economic and environmental benefits by reducing unnecessary fuel consumption through route optimization. It can also prevent catastrophic losses, such as oil spills and casualties, by detecting risks such as collisions between ships and between ships and marine structures in advance. Therefore, studies

*Corresponding author: e-mail: mhjeong@chosun.ac.kr
**Corresponding author: e-mail: dooyongcho@cnu.ac.kr
†These authors contributed equally to this work.
https://doi.org/10.18494/SAM4437

related to abnormal behavior detection and vessel traffic prediction have been performed to achieve a safer and more efficient maritime traffic network.[1–3]

The automatic identification system (AIS) data collected by navigation equipment that can automatically transmit and receive information such as ship identification or location contribute significantly to the development of advanced VTSs.[4–6] However, extracting meaningful features from extensive data to produce the desired results is challenging. Moreover, the data collected in the real world include various errors and outliers, which can degrade the performance of the defined model. For example, the non-normal distribution of data can distort the view of the data, leading to the poor performance of traditional statistical methods. In addition, if the data are nonlinear, it is challenging to perform predictive tasks with models that cannot handle nonlinearity because conventional methods assume linearity and have a simple structure.

Deep learning models alleviate these problems and have recently achieved good performance in several fields.[7,8] Thus, deep-learning-based models have been actively investigated for vessel trajectory prediction since 2018.[9] In this study, among the various deep learning methods, long short-term memory (LSTM) was used for vessel trajectory prediction. LSTM performs well in trajectory prediction and is one of the most widely used deep learning methods. Moreover, many studies have been performed on the extension of LSTM.

However, in previous studies, a loss function that minimized the distance between the destination (target) and the predicted location (output point) was used to train the deep learning model. Such a loss function has difficulty dealing with various elements. Therefore, in this paper, we present a novel loss function to minimize the area of the origin, destination, and predicted location. The performance of the proposed method was compared with that of the state-of-the-art method.

The remainder of this paper is organized as follows. In Related Works, we review previous studies on vessel trajectory prediction to confirm the necessity and contribution of the present study. In Materials and Methods, we detail the AIS data and LSTM models used in the experiments, as well as the loss function, and present the metrics used for model evaluation. In Results, the proposed loss function is compared with the conventional loss function, and the performance improvement is verified. The results and contributions of this study are summarized in Discussion and Conclusions. Additionally, the limitations of the study and directions for future research are discussed.

2. Related Works

The methods used for vessel trajectory prediction can be largely divided into statistical, machine learning, and deep learning methods. The statistical methods [i.e., autoregressive integrated moving average (ARIMA), Gaussian processes, and Markov chain] have been applied to tasks such as vessel trajectory prediction.[10–12] Such statistical methods have the advantages of simple formulas and easy calculations, but have various limitations. For example, ARIMA has an assumption of stationarity; however, the data collected, along with the data distribution required by the parametric approach, typically fail to satisfy this assumption.

In the case of the Kalman filter,[13] it is assumed that the error in the data follows a normal distribution, and that the current and previous states have a linear relationship. Furthermore, these models are inadequate for large amounts of data.[14]

The amount of research on the use of deep learning for vessel trajectory prediction is increasing. With the development of sensors, deep learning methods that can make predictions on the basis of data have attracted considerable attention. Furthermore, deep learning, which builds layers for nonlinearity, outperforms other methods in many practical applications. It is suitable for application to real-world data because it does not assume the distribution of the data.

Among the deep learning models used for vessel traffic prediction, the artificial natural network (ANN) is the simplest. This algorithm trains the model using backpropagation, inspired by the central nervous systems of animals—particularly the brain. ANNs have been applied to vessel trajectory prediction.[15] However, because of the simple structures of ANNs, they have limitations for considering spatial and temporal features that can affect the vessel trajectory. Researchers have proposed various deep learning models to address these limitations.

Graph convolutional networks (GCNs) are widely used for trajectory prediction.[16–18] They can reflect non-Euclidean information, such as relationships. However, they cannot capture time-series characteristics. Thus, the spatiotemporal graph convolutional network has been used for vessel trajectory prediction, which combines information from various graph-based models.[19]

The recurrent neural network (RNN) is suitable for handling time-series data, and its variations have become the most widely used prediction methods. In particular, LSTM is widely used to alleviate the vanishing gradient problem of traditional RNN-based models, and it is used in trajectory prediction with various extensions.[20–22]

While the aforementioned studies focused on deep learning models, in another study,[23] a new loss function was developed to improve the model performance. The existing loss function minimizes the Euclidean distance between the target and output points generated by the model.

In addition to the distance between the target and output points, the loss function of the study considers the angle among the target, origin (input), and output points. Similarly, in this study, the area of the target, origin (input), and output points were considered in the loss function to train and improve the model. The LSTM model was used for vessel trajectory prediction. LSTM has been widely applied in various fields.[24–27] Moreover, the study of the previous loss function used LSTM for prediction. Therefore, the proposed loss function was applied to the LSTM model to improve the performance and perform the comparison in the present study.

3. Materials and Methods

3.1 Data

In this study, AIS data from navigation equipment were used. The data were collected from January 1 to December 31, 2020, and came from approximately 28000 ships that sailed around South Korea, including cargo, tanker, medical, fishing, and cruise ships. Figure 1 shows a sample of the data.

Fig. 1. (Color online) Visualization of AIS data.

In this study, 100 of the approximately 28000 ships were randomly extracted. Then, we defined models and conducted the prediction when the sequence length of the trajectory was ≥500 because a small sequence length cannot be used to train models. Thus, models for 53 of the 100 vessels were defined.

3.2 Long short-term memory

In this study, the LSTM model was used to predict the vessel trajectory. Figure 2 shows the structure of LSTM.[28] LSTM was proposed to alleviate the vanishing gradient problem of existing RNNs. It updates the cell state using three gates, namely, forget, input, and output, which employ the following formulas:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \tag{1}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \tag{2}$$

$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right). \tag{3}$$

Here, f, i, and o represent the forget, input, and output gates, respectively. The hidden state is expressed as $h_t = o_t * \tanh(C_t)$, and σ represents a sigmoid activation function. LSTM manages the cell state by opening and closing the three gates described by Eqs. (1)–(3), which can alleviate the vanishing gradient problem. In LSTM, the cell state is defined as

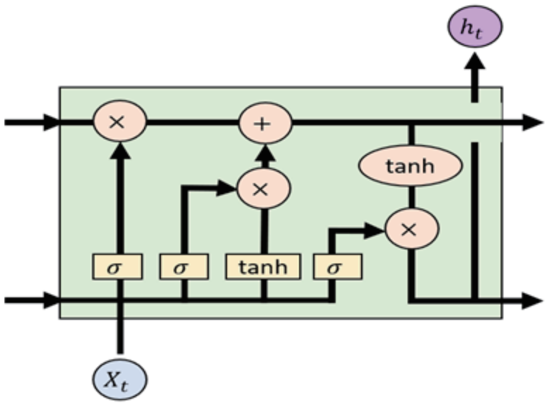

Fig. 2. (Color online) LSTM architecture.

$$\widetilde{C_t} = \tanh\left(W_C \cdot \left[h_{t-1},\ x_t\right] + b_C\right), \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}. \tag{5}$$

3.3 Loss function

The loss function is a crucial component of deep learning. It calculates the model's output on the basis of the weights and activation function, takes in actual data as input, and measures the difference between the model's output and the desired targets. The loss function is used to optimize the weights until it can accurately predict the target variable. In regression problems, the goal is to predict a continuous output variable on the basis of one or more input variables.

The most common loss function for regression is the mean squared error (*MSE*), which measures the average squared difference between the predicted and actual values. Another popular loss function for regression is the mean absolute error (*MAE*), which measures the average absolute difference between the predicted and actual values. Both loss functions are used to optimize the parameters of a regression model during training. The goal is to minimize the loss function by adjusting the model's weights and biases until the predicted values are as close as possible to the actual values.

We developed a loss function considering a two-dimensional area to improve the performance of the deep learning model for trajectory prediction and compared it with a loss function considering distance and direction.

Figure 3 shows the input (A), ground truth (B), and output (C) points. Let $\Delta_n^{u,v} = \left(x_n^u - x_n^v\right)^2 + \left(y_n^u - y_n^v\right)^2$, with $u, v \in \{A, B, C\}$. The loss function that was most commonly used in previous studies, which only considers the Euclidean distance between the output and ground truth, is

$$loss_1 = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\Delta_n^{B,C}}, \tag{6}$$

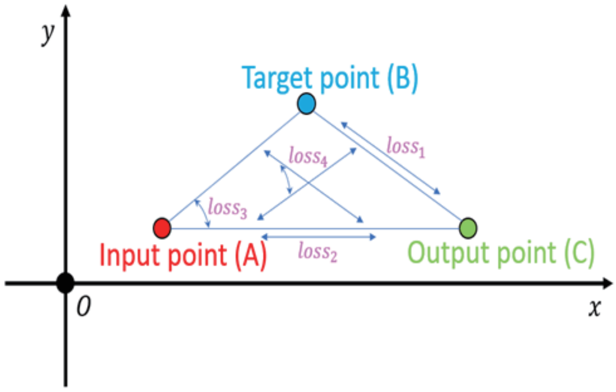

Fig. 3. (Color online) Relationships between the different points and the loss function.

where N represents the number of vessel trajectories. In one study, Eqs. (7) and (8) were added.[23] Equation (7) is a loss function for learning how far output data should be from input data. This is independent of direction but is an important factor to consider for accuracy.

$$loss_2 = \frac{1}{N}\sum_{n=1}^{N}\left(\sqrt{\Delta_n^{A,B}} - \sqrt{\Delta_n^{A,C}}\right)^2 \tag{7}$$

$$loss_3 = \frac{1}{N}\sum_{n=1}^{N}\arccos\left(\frac{\Delta_n^{A,B} + \Delta_n^{A,C} - \Delta_n^{B,C}}{2\times\sqrt{\Delta_n^{A,B}}\times\sqrt{\Delta_n^{A,C}}}\right)^2 \tag{8}$$

Equation (8) is the second law of cosines. This is a loss function for ensuring the convergence of the angle BAC to zero. The proposed loss function, which aims at the convergence of an area of three points to zero, is given as

$$loss_4 = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\sqrt{\Delta_n^{A,B}}\sqrt{\Delta_n^{A,C}}\sin(\alpha)^2, \tag{9}$$

where α represents the angle of the triangle formed by the three points. In our experiment, the model was defined, and statistical tests were performed to compare the performance of the proposed loss function with that of the previous loss function for vessel trajectory prediction. The previous and proposed loss functions are given by Eqs. (10) and (11), respectively.

$$previous\ loss\ function = loss_1 + loss_2 + loss_3 \tag{10}$$

$$proposed\ loss\ function = loss_1 + loss_4 \tag{11}$$

3.4 Evaluations

MAE and R^2 were used to evaluate the model performance. *MAE* is a statistical measure of errors between two observations reflecting the same phenomena. R^2 is a statistical measure that quantifies the proportion of the variance explained by an independent variable in a regression model for a dependent variable. They are defined as

$$MAE = \sum_{i=1}^{n} \left| y_i - x_i \right|, \tag{12}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(x_i - y_i \right)^2}{\sum_{i=1}^{n} \left(x_i - \overline{y_i} \right)}, \tag{13}$$

where i represents the number of samples in the test set of each model, x represents the model output, and y represents the ground truth. In addition, the percent difference was used to quantify differences between models. It is defined as

$$percent\ difference = \frac{\left| V_1 - V_2 \right|}{\left[\frac{\left(V_1 + V_2 \right)}{2} \right]} \times 100, \tag{14}$$

where V_1 and V_2 represent the performance characteristics of two models given by metrics such as *MAE* and R^2.

4. Results

In our experiment, the predictive performance characteristics of the two models with different loss functions were compared and analyzed, and the results are shown in Table 1. The performance characteristics were evaluated by defining a model for the 53 vessels considering only the sequence length of 500.

As indicated by the average values of *MAE* and R^2, the proposed loss function can make vessel trajectory prediction more accurate and robust. The proposed loss function outperformed the previous loss function. A comparison of the percent differences revealed 12.77 and 0.47% for *MAE* and R^2, respectively.

In addition, a statistical test was conducted on the *MAE* and R^2 values of the models to determine whether the differences in the results were statistically significant. Table 2 presents the statistical test results and the percent difference of means. A t-test is used to determine whether the difference between two groups is significant. If the p-value is less than the significance level (0.05), the difference between two groups is significant. Although there was no significant difference in *MAE*, the t-test confirmed that the difference in R^2 was statistically significant.

Moreover, Fig. 4 shows the predicted output and actual data for a single vessel from samples on a map. By comparing the proposed loss function with the previous one, it is apparent that it produces more accurate results closer to the actual data upon visual inspection.

Table 1
Results of experiments.

Index	Previous loss function		Proposed loss function	
	MAE	R^2	*MAE*	R^2
1	0.001244	0.951050	0.001280	0.943543
2	0.019567	0.914244	0.019732	0.919088
3	0.003186	0.980985	0.003415	0.976383
4	0.055157	0.911978	0.033388	0.960052
5	0.014349	0.984916	0.007315	0.994115
6	0.010199	0.972401	0.010004	0.973072
7	0.020175	0.965403	0.020202	0.965309
8	0.003466	0.993068	0.005561	0.992492
9	0.051834	0.934257	0.024249	0.982669
10	0.019963	0.973741	0.018725	0.974467
11	0.022571	0.983869	0.023633	0.982141
12	0.014127	0.995563	0.013370	0.995872
13	0.020456	0.983519	0.005554	0.997512
14	0.016167	0.955367	0.016191	0.955648
15	0.025620	0.967005	0.027027	0.968692
16	0.010083	0.949495	0.010259	0.946190
17	0.015925	0.953033	0.015890	0.953203
18	0.005253	0.999645	0.005415	0.999580
19	0.010843	0.960419	0.010465	0.962004
20	0.004554	0.971441	0.005014	0.982240
21	0.014003	0.995448	0.012081	0.995893
22	0.004097	0.999205	0.004058	0.999073
23	0.005223	0.999325	0.004808	0.999493
24	0.003075	0.999037	0.003758	0.998703
25	0.003667	0.999128	0.004647	0.998087
26	0.004975	0.996965	0.005239	0.995768
27	0.008698	0.996758	0.011073	0.986425
28	0.004574	0.998053	0.003759	0.999113
29	0.018938	0.975151	0.010644	0.991719
30	0.003287	0.992411	0.002445	0.989021
31	0.007573	0.990580	0.004943	0.997188
32	0.007964	0.994868	0.010396	0.992768
33	0.003961	0.999214	0.009965	0.996824
34	0.005829	0.993450	0.006562	0.991846
35	0.003082	0.916095	0.002775	0.985271
36	0.002644	0.999494	0.002513	0.999485
37	0.003947	0.999058	0.003491	0.999417
38	0.003350	0.996702	0.005404	0.995429
39	0.003564	0.999029	0.003730	0.999115
40	0.017444	0.926482	0.005390	0.994930
41	0.004814	0.951732	0.004678	0.953642
42	0.003197	0.964527	0.004693	0.962380
43	0.001317	0.998357	0.001306	0.997741
44	0.016294	0.963319	0.015696	0.963353
45	0.017246	0.979846	0.017802	0.978491
46	0.004687	0.993855	0.005205	0.995701
47	0.006334	0.999332	0.005827	0.999408
48	0.012632	0.939559	0.015013	0.927763
49	0.004422	0.999572	0.005611	0.999205
50	0.009195	0.938940	0.008395	0.937297
51	0.006486	0.998554	0.011774	0.996258
52	0.016661	0.993514	0.016167	0.993224
53	0.004665	0.999581	0.005725	0.999477

Table 2
Statistical results.

Metrics	*MAE*	R^2
Percent difference of means	12.77%	0.47%
t-test	p-value > 0.05	p-value < 0.05

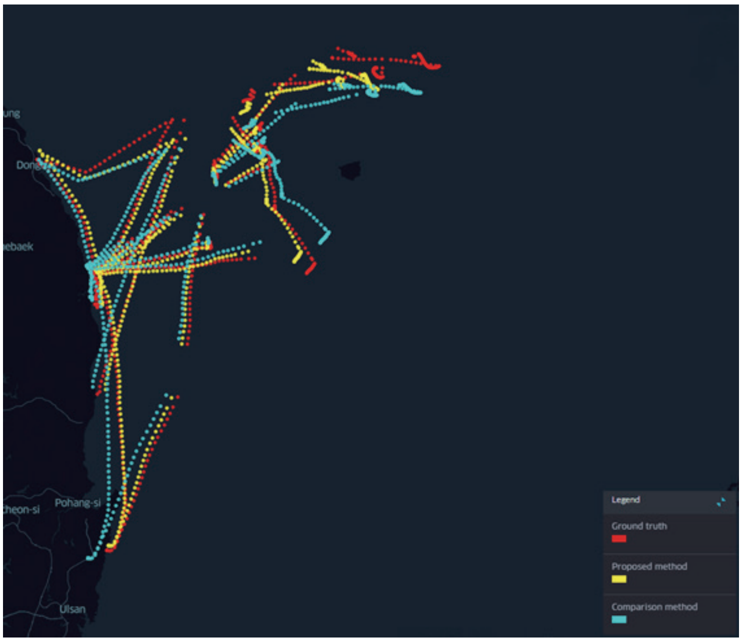

Fig. 4. (Color online) Visualization of prediction results.

6. Discussion and Conclusions

In this paper, we presented a new loss function for vessel trajectory prediction based on AIS data. The proposed loss function improves the model performance by considering both the Euclidean distance and the area. The proposed loss function outperformed a previously reported loss function in our experiments. In particular, the *t*-test confirmed that the difference in R^2 was statistically significant. Furthermore, the percent difference in means revealed that the proposed loss function performed better than the alternative loss function. The *MAE* and R^2 values differed by approximately 12.07 and 0.47%, respectively.

In training a deep learning model, the loss function measures the difference between outputs and targets. Furthermore, the training process updates the parameters through backpropagation on the error calculated using a loss function. Thus, the loss function is one of the vital factors of the deep learning structure. In previous studies, the loss function that minimizes the Euclidean distance for vessel trajectory prediction was widely used. However, it is necessary to develop alternative loss functions to improve the performance of the deep learning models. The results of this study can contribute to the optimization of VTSs.

However, we did not apply the multiple data previously used, such as relationships or correlation. Moreover, we used the data generated by 53 ships by defining a model only when the sequence length was above a certain level among randomly extracted samples. Future studies should focus on model generalization by considering vessels' size and weather.

Acknowledgments

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2021R1C1C1012785).

References

1 M. Stasolla, J. J. Mallorqui, G. Margarit, C. Santamaria, and N. Walker: IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **9** (2016) 6. https://doi.org/10.1109/JSTARS.2016.2551730
2 M. Tello, C. López-Martínez, and J. J. Mallorqui: IEEE Geosci. Remote Sens. Lett. **2** (2005) 5. https://doi.org/10.1109/LGRS.2005.845033
3 S. Hexeberg, A. L. Flåten, and E. F Brekke: Proc. 2017 20th Int. Conf. Information Fusion (Fusion) (IEEE, 2017) 1–8.
4 X. Han, C. Armenakis, and M. Jadidi: Sustainability **13** (2021) 15. https://doi.org/10.3390/su13158162
5 L. Zhao, G. Shi, and J. Yang: J. Navig. **75** (2018) 5. https://doi.org/10.1017/S0373463318000188
6 D. W. Gao, Y. S. Zhu, J. F. Zhang, Y. K. He, K. Yan, and B. R. Yan R: Ocean Eng. **228** (2021) 108956. https://doi.org/10.1016/j.oceaneng.2021.108956
7 S. B. Jeon, M. H. Jeong, T. Y. Lee, J. H. Lee, and J. M. Cho: Sens. Mater. **32** (2020) 4441. https://doi.org/10.18494/SAM.2020.3111
8 M. S. Khan, S. B. Jeon, and M. H. Jeong: Remote Sens. **13** (2021) 24. https://doi.org/10.3390/rs13244976
9 X. Zhang, X. Fu, Z. Xiao, H. Xu, and Z. Qin: IEEE Trans. Intell. Transp. Syst. **23** (2022) 11. https://doi.org/10.1109/TITS.2022.3192574
10 M. S. Ahmed and A. R. Cook: Transp. Res. Rec. **772** (1979) 1.
11 H. Rong, A. P. Teixeira, and C. G. Soares: Ocean Eng. **182** (2019) 499. https://doi.org/10.1016/j.oceaneng.2019.04.024
12 S. Guo, C. Liu, Z. Guo, Y. Feng, F. Hong, and H. Huang: Proc. Wireless Algorithms, Systems, and Applications: 13th Int. Conf. (Springer International Publishing, 2018) 13.
13 A. Stathopoulos and M. G. Karlaftis: Transp. Res. Part C Emerging Technol. **11** (2003) 2. https://doi.org/10.1016/S0968-090X(03)00004-4
14 N. Meng, E. Y. Lam, K. K. Tsia, and H. K. So: IEEE J. Biomed. Health. Inf. **23** (2018) 5. https://doi.org/10.1109/JBHI.2018.2878878
15 U. Simsir and S. Ertugrul: Appl. Soft Comput. **9** (2009) 4. https://doi.org/10.1016/j.asoc.2009.03.002
16 X. Li, X. Ying, and M. C. Chuah: Proc. 2019 IEEE Intelligent Transportation Systems Conf. (IEEE, 2019) 3960–3966.
17 L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (IEEE, 2021) 8994–9003.
18 A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (IEEE, 2020) 14424–14432.
19 M. Liang, R. W. Liu, Y. Zhan, H. Li, F. Zhu, and F. Y. Wang: IEEE Trans. Intell. Transp. Syst. **23** (2022) 12. https://doi.org/10.1109/TITS.2022.3199160
20 A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2016). 961–971.
21 Y. Suo, W. Chen, C. Claramunt, and S. Yang: Sensors **20** (2020) 18. https://doi.org/10.3390/s20185133
22 H. Tang, Y. Yin, and H. Shen: J. Mar. Eng. Technol. **21** (2022) 3. https://doi.org/10.1080/20464177.2019.1665258
23 R. W. Liu, M. Liang, J. Nie, W. Y. B. Lim, Y. Zhang, and M. Guizani: IEEE Trans. Network Sci. Eng. **9** (2022) 5. https://doi.org/10.1109/TNSE.2022.3140529
24 B. Murray and L. P. Perera: Reliab. Eng. Syst. Saf. **215** (2021) 107819. https://doi.org/10.1016/j.ress.2021.107819
25 W. Li, C. Zhang, J. Ma, and C. Jia: Proc. 2019 5th Int. Conf. Transportation Information and Safety (IEEE, 2019) 1389–1394.

26 J. Venskus, P. Treigys, and J. Markevičiūtė: Nonlinear Anal.-Model. Control **26** (2021) 4. https://doi.org/10.15388/namc.2021.26.23056
27 R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (IEEE, 2019) 8483–8492.
28 S. Hochreiter and J. Schmidhuber: Neural Comput. **9** (1997) 8. https://doi.org/10.1162/neco.1997.9.8.1735