



RESEARCH ARTICLE

# A prediction model of vessel trajectory based on generative adversarial network

Senjie Wang,<sup>1</sup> and Zhengwei He<sup>1,2,3\*</sup>

<sup>1</sup> School of Navigation, Wuhan University of Technology, Wuhan, China.

<sup>2</sup> Key Laboratory of Hubei Province for Inland Navigation Technology, Wuhan University of Technology, Wuhan, China.

<sup>3</sup> National Engineering Research Center for Water Transportation Safety, Wuhan University of Technology, Wuhan, China.

\*Corresponding author. E-mail: [wwwzhzw@whut.edu.cn](mailto:wwwzhzw@whut.edu.cn)

Received: 8 July 2020; Accepted: 3 April 2021; First published online: 26 April 2021

**Keywords:** trajectory prediction, ship encounter, generative adversarial network, deep learning

## Abstract

Trajectory prediction is an important support for analysing the vessel motion behaviour, judging the vessel traffic risk and collision avoidance route planning of intelligent ships. To improve the accuracy of trajectory prediction in complex situations, a Generative Adversarial Network with Attention Module and Interaction Module (GAN-AI) is proposed to predict the trajectories of multiple vessels. Firstly, GAN-AI can infer all vessels' future trajectories simultaneously when in the same local area. Secondly, GAN-AI is based on adversarial architecture and trained by competition for better convergence. Thirdly, an interactive module is designed to extract the group motion features of the multiple vessels, to achieve better performance at the ship encounter situations. GAN-AI has been tested on the historical trajectory data of Zhoushan port in China; the experimental results show that the GAN-AI model improves the prediction accuracy by 20%, 24% and 72% compared with sequence to sequence (seq2seq), plain GAN, and the Kalman model. It is of great significance to improve the safety management level of the vessel traffic service system and judge the degree of ship traffic risk.

## 1. Introduction

Maritime transportation accounts for around 90% of world trade, and the trade volume is still growing at a rate even faster than the global economy (Kaluza et al., 2010). With the increase in numbers of ships in the world comes an increase in the necessity for safety, intelligent shipping and efficiency. As a fundamental research, trajectory prediction is a key technology to support accident early warning and path planning without collision for intelligent navigation (Borkowski, 2010; Tu et al., 2017). The ship trajectory prediction model with high prediction accuracy and good stability is of great significance to ensure the safety of transportation and to promote the development of intelligent shipping.

At present, there are two main kinds of trajectory prediction methods: one is based on a kinematics model while another is based on a neural network model. The methods based on the kinematics model must build the state transfer equation to describe the motion law (Sutulo et al., 2002). Krzysztof (2017) used the discrete Kalman filter to estimate the missing vessel trajectory sequence. Perera and Soares (2010) used the extended Kalman filter to estimate the future position of the vessel by establishing the state transition matrix. Zhang et al. (2019) and Huang (2018) established the state transition model based on the hidden Markov model, and estimated the ship position in the future through the Markov chain. Liu et al. (2019a) fused multiple source of sensors to improve the Markov method's accuracy, and the average error of prediction is lower than 5 km in the future 5 h. These cited methods usually have the following disadvantages: first, the kinematics model is difficult to establish, as a result of considering

the hydrological environment factors such as water and wind. Second, these methods cannot model the ship's continuous motion behaviour. As a result, these methods are not suited to predict future positions at continuous time points. In recent years, with the development of machine learning and deep learning, the artificial neural network model has become a research hotspot. Borkowski (2017) fused multiple-source data on the basis of automatic identification system (AIS) data and obtained higher prediction accuracy by using a long-short-term memory (LSTM) model. The average error of prediction is lower than 78 m in the future 6 min. Tang et al. (Tang et al., 2019) used an LSTM neural network to model the ship's trajectory, mined the temporal and spatial distribution of the trajectory through data-driven method and predicted the ship's future trajectory. Their work achieved an average error of prediction that was less than 280 m in the future 10 min. Li et al. (Li et al., 2019) utilised the clustering method to capture the trajectories' features firstly and then used the LSTM to achieve an average error of prediction that was less than 50 m in the future 10 min. Liu et al. (2019b) improved the prediction performance of the plain Support Vector Machine (SVM) model through the adaptive chaos differential evolution algorithm to optimise the model parameters. The average error of prediction is less than 33.41 m in the future 60 s. Suo et al. (Suo et al., 2020) used the Gate Recurrent Unit Neural Network (GRU) to improve computational time efficiency while the prediction accuracy is similar to LSTM. Although the existing methods have made great progress in vessel trajectory prediction, there are still limitations in two aspects. On the one hand, these methods do not consider the vessel's interaction in a narrow complex traffic environment. In practice, the behaviour of the vessel is not only related to its own state but is also related to other vessels' positions in the same water area. These works have achieved good results in the prediction of single-ship trajectory in free water areas, but when the target encounters with other vessels, the performance of this method will be poor, due to the fact that it ignores any interaction information. On the other hand, the current method often uses the Euclidean distance between the real trajectory and the predicted trajectory as the loss function; it makes the model tend to learn the 'average behaviour' of the training data and ignores the vessel's 'avoid behaviour' at the true encounter.

In this paper, the Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is used to model the ship trajectory prediction task. To model ship interaction behaviour, the motion features are divided into self-motion features and group motion features. The self-motion features include the single vessel's speed, acceleration, location and other data. The group motion features include the multiple vessels' relative speed, relative acceleration and other data. In the generator of GAN-AI, an encoder based on LSTM is used to process and extract the self-motion features. An interactive module based on another LSTM is designed to process and analyse the group motion features. In the encoder and decoder, the attention mechanism (Luong et al., 2015) is utilised to improve the model's ability to model the temporal-spatial sequence reasonably. Then the Multilayer Perceptron (MLP) (Sainath et al., 2015) is used to reshape the data and output the future trajectory. Lastly, the network is optimised and trained by the relative adversary loss and the variety loss of generator during training, to achieve a higher precision result.

## 2. Problem definition

This paper aims to use the longitude  $x$ , latitude  $y$ , speed  $v$ , course  $\theta$  recorded by AIS (Sun and Zhou, 2017) data to jointly infer the future trajectory of all ships involved in a ship encounter. To facilitate the processing of vector data, the velocity of the vessel  $i$  at the time  $t$  is decomposed into the decomposition speed in the X and Y directions:

$$v_{xi}^t = v_i^t \cdot \sin \theta_i^t, t \in [1, t_{obs}] \quad (1)$$

$$v_{yi}^t = v_i^t \cdot \cos \theta_i^t, t \in [1, t_{obs}] \quad (2)$$

It assumes that each trajectory sequence  $X_i$  is composed of discrete trajectory sampling points  $\mathbf{p}_i^t$ . Each discrete sampling point  $\mathbf{p}_i^t$  includes the vessel's coordinate data and decomposition speed in the X

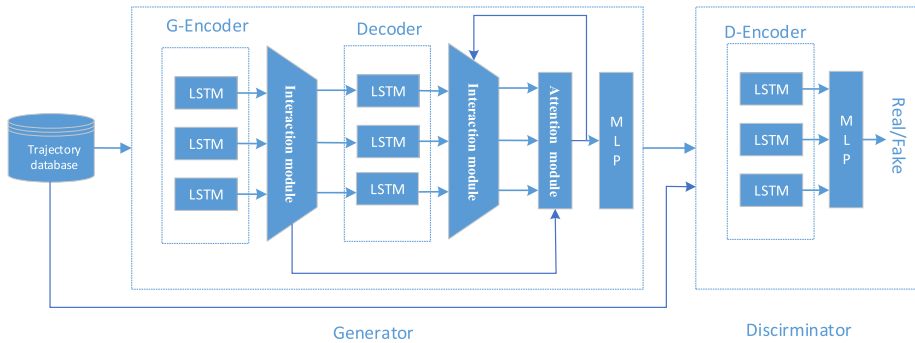


Figure 1. Structure of GAN-AI.

and Y directions:

$$\mathbf{p}_i^t = (x_i^t, y_i^t, v_{xi}^t, v_{yi}^t) \tag{3}$$

$$\mathbf{X}_i = (\mathbf{p}_i^1, \mathbf{p}_i^2, \dots, \mathbf{p}_i^{\text{obs}}) \tag{4}$$

If the input  $\mathbf{X}$  is the matrix including all ships' trajectory sequence which at the same water area:

$$\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n) \tag{5}$$

The task of the GAN-AI is to predict the future trajectory of all ships at the same time:

$$\hat{\mathbf{Y}} = (\hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}_2, \dots, \hat{\mathbf{Y}}_n) \tag{6}$$

$$\hat{\mathbf{Y}}_i = ((x_i^{\text{obs}+1}, y_i^{\text{obs}+1}), (x_i^{\text{obs}+2}, y_i^{\text{obs}+2}), \dots, (x_i^{\text{pred}}, y_i^{\text{pred}})) \tag{7}$$

To help the model better learn the ship interaction, the GAN-AI model uses asymmetric input and output. The input data includes the ship position information and velocity information. In the encoder, the interaction module processes the single ship information into the relative distance information and relative speed information between the multiple vessels, to learn the vessels' interaction behaviour and better estimate the future trajectory. The output data only includes the coordinate information of the future vessel. It can make the model focus on the ship position prediction and improves the network convergence ability by reducing the output variables of the multi-output neural network.

### 3. GAN-AI model

The structure of GAN-AI model is shown in Figure 1. The model consists of a generator and discriminator. The generator consists of a G-encoder, G-decoder, interaction module, attention module and MLP. The discriminator consists of a D-encoder and MLP. During the training, the vessels' historical trajectory sequence  $\mathbf{X}$  is encoded by the G-encoder, and the coding matrix of each trajectory  $H_e$  is obtained. Through the interaction module, the integrated data  $C$  that includes single information and group information is collected. The G-decoder decodes the  $C$  as the context information to output the predicted trajectory  $\hat{\mathbf{Y}}$ . The real trajectory and generated trajectory are input into the discriminator to distinguish the reality of the trajectory, and the network parameters of the generator and discriminator are optimised to make the predicted trajectory approach the real trajectory by decreasing the model loss.

#### 3.1. Interaction model

An interactive module is used to process and learn the encounter information of multiple ships in the same water area. At time  $t$ , the interaction module calculates the difference of coordinates and speeds

between the ship  $i$  and  $j$  to get the relative distance information and relative speed information. Then coding the relative motion information through a fully connected neural network (FCNN) (Sainath et al., 2015) to get the hidden interaction information  $s_{ij}^t$ , and then stores the group interaction information into a set as  $S^t$ . Finally, the individual motion information  $H_e^t$  and the group interaction information  $S^t$  are together input into a MLP to obtain integrated state vector. The relevant formulae are:

$$C^t = \text{Interact}(\mathbf{X}, E) \Rightarrow \begin{cases} s_{ij}^t = FC(x_i^t - x_j^t, y_i^t - y_j^t, v_{xi}^t - v_{xj}^t, v_{yi}^t - v_{yj}^t; W_I) \\ S^t = [s_{12}^t, \dots, s_{ij}^t, \dots, s_{N-1N}^t], (i \neq j) \\ C^t = MLP([S^t, H_e^t]; W_{\text{tanh}}) \end{cases} \quad (8)$$

where  $W_I, W_{\text{tanh}}$  is the weight matrices of the FCNN and the MLP with tanh activation function that must be trained.  $H_e^t$  is the coding data obtained from the G-encoder and  $H_e^t$  includes the self-motion features of all ships at time  $t$ .

### 3.2. Generator

The generator is mainly composed of a G-encoder and G-decoder, which are used to capture and analyse the semantic information of historical trajectory and generate reasonable future trajectory sequence. In the G-encoder, each trajectory sequence  $\mathbf{X}_i$  is mapped and normalised by an FCNN to obtain vector  $E_i$ , and then the  $E_i$  is encoded by an LSTM (Bahdanau et al., 2014) to obtain the hidden representation of trajectory  $H_{ei}$ :

$$E_i = FC(\mathbf{X}_i; W_{em}) \quad (9)$$

$$H_{ei} = LSTM(E_i, H_{init}; W_{en}) \quad (10)$$

where  $W_{em}$  and  $W_{en}$  are the weight matrices of the FCNN and the LSTM that must be trained.  $H_{ei}$  is the output of the LSTM neural network, which involves the potential characteristic information of the position, speed, acceleration and so on of vessel  $i$ .  $H_{init}$  is the initial state of the LSTM's hidden layer which is set by normal distribution. Through  $N$  times of processing, we can get the code matrix  $H_e$  that includes the self-motion features of  $N$  ships in the same water area:

$$H_e = [H_1, H_2, \dots, H_N] \quad (11)$$

To model the interactive behavior of the multiple ships, we input the  $H_e$  and the  $X$  into the interaction module introduced in Section 3.1. The interaction module can integrate the group motion information and the monomer motion information at time  $t_{obs}$ . The output  $C$  of interaction module contains the  $N$  ships' own motion information and relative motion information:

$$C = [C_1, \dots, C_i, \dots, C_N] = \text{Interact}(\mathbf{X}, H_e) \quad (12)$$

To predict the future trajectory point of vessel  $i$  at time  $t$ , the G-decoder must be input the coordinate data  $(x_i^{t-1}, y_i^{t-1})$  and context information  $C_i^{t-1}$  that were obtained from the last time point, and then the G-decoder can predict the trajectory point coordinates on continuous time points by iteration. At the beginning of prediction, Gaussian noise  $z$  is superimposed  $C_i$  to get  $C_i'$ , and the initial hidden state of the LSTM decoder is initialised by  $C_i'$  to improve the noise reduction ability of the decoder:

$$C_i' = [C_i, z] \quad (13)$$

$$e_{di}^t = FC(x_i^{t-1}, y_i^{t-1}; W_{ed}) \quad (14)$$

$$h_{di}^t = LSTM[e_{di}^t, C_i^{t-1}; W_{de}] \quad (15)$$

where  $W_{ed}, W_{de}$  is the network matrices of the FCNN and LSTM that must be trained, and  $h_{di}^t$  is the output LSTM. The FCNN in Equation (14) is used to map the coordinate to feature space and the LSTM is used to decode context  $C_i^{t-1}$ . To solve the problem of memory degradation in the iterative updating process of LSTM and make the model always focus on the real observation data  $C_i^t$  from the G-encoder,  $C_i^t$  and  $h_{di}^t$  are concatenated to calculate the attention weights  $a_i^t$  through the attention module at each update step. The weighted  $h_{di}^t$  can be obtained by multiplying  $h_{di}^t$  and  $a_i^t$ , and then it is employed to achieve coordinate prediction with an MLP. The formula is:

$$a_i^t = FC([h_{di}^t, C_i^t]; W_a) \tag{16}$$

$$(\hat{x}_i^t, \hat{y}_i^t) = MLP(a_i^t \cdot h_{di}^t; W_{relu}) \tag{17}$$

where  $W_d, W_{relu}$  is the network matrices of the FCNN and MLP that must be trained. The FCNN in Equation (16) is used to calculate the similarity between the  $h_{di}^t$  and  $C_i^t$  to get the attention weights. The MLP is used to shape the vector and output the future position. Finally, the prediction results are input into the interaction module to update the context information  $C_i^t$ , and the  $C_i^t$  can be used to prediction for next time step.

### 3.3. Discriminator

The discriminator is a classical binary classification model to judge whether the generated trajectory is true or false. The discriminator and the generator compete with each other to make themselves converge at the same time during training. The discriminator maps the trajectory sequences through an FCNN to get feature vector  $R$ . An LSTM is used to further encode the  $R$  to get  $H_c$ . Then run  $H_c$  through an MLP to output the reality of the trajectories:

$$R = FC(M; W_{ec}), M \in \{X_r, X_f\} \tag{18}$$

$$H_c = LSTM(h_{init}, R; W_c) \tag{19}$$

$$Q = MLP(H_c; W_s) \tag{20}$$

where,  $W_{ec}, W_c, W_s$  are the parameters of the FCNN, LSTM, and MLP that must be trained.  $X_r$  is the set of real trajectory sequence,  $X_f$  is the set of fake trajectory sequence,  $h_{init}$  is the initialisation state parameter of the LSTM network,  $H_c$  is the output of the LSTM.

### 3.4. Loss function

The relative average loss (Wang et al., 2018) is used as the metric to estimate the distribution differences between the fake and the real trajectories. The relative average loss function is defined as the probability that the real trajectory is more realistic than the predicted trajectory in the discriminator. The formulae are:

$$D_{Ra}(x_r, x_f) = \sigma(Q(x_r) - E[Q(x_f)]) \tag{21}$$

$$D_{Ra}(x_f, x_r) = \sigma(Q(x_f) - E[Q(x_r)]) \tag{22}$$

where  $x_r$  denotes the real trajectory sequence,  $x_f$  denotes the predicted trajectory sequence, and  $\sigma$  is a sigmoid function. In the GAN-AI model, the discriminator is only used to estimate the probability that the input is the real trajectory, so the loss function of the discriminator is defined as:

$$L_D = -E_{x_r}[\log(D_{Ra}(x_a, x_f))] - E_{x_f}[1 - \log(D_{Ra}(x_f, x_r))] \tag{23}$$

The generator loss function is composed of  $L_G^{Ra}$  and  $L_{L2}$ ,  $L_G^{Ra}$  is the adversarial loss function of the generator and the discriminator and realises the competition between the discriminator and the

generator.  $L_{L2}$  is the variety loss (Gupta et al., 2018) to encourage GAN-AI to produce diverse samples. It means the generator generates  $k$  trajectories during training and chooses the minimum loss of the real trajectory and the predicted trajectories as the model’s loss. The formulae are:

$$L = \arg \min_G \max_D (L_G^{Ra} + L_{L2}) \tag{24}$$

$$L_G^{Ra} = -E_{x_r} [\log(1 - D_{Ra}(x_r, x_f))] - E_{x_f} [\log(D_{Ra}(x_f, x_r))] \tag{25}$$

$$L_{L2} = \min_k E [\|x_r - x_f\|_2] \tag{26}$$

**4. Experiment**

To assess the correctness and accuracy of the GAN-AI, the historical AIS data of Zhou Shan Port, China, in January 2018 was selected to evaluate the model. That part of the channel is an intersection that is not only narrow but also has high traffic density. A large number of small and medium-sized ships with north-south and east-west directions encounter here every day, causing a complex traffic environment. The experimental data are 2600 ship trajectories, including 2000 trajectories as training data and 600 trajectories as validation data.

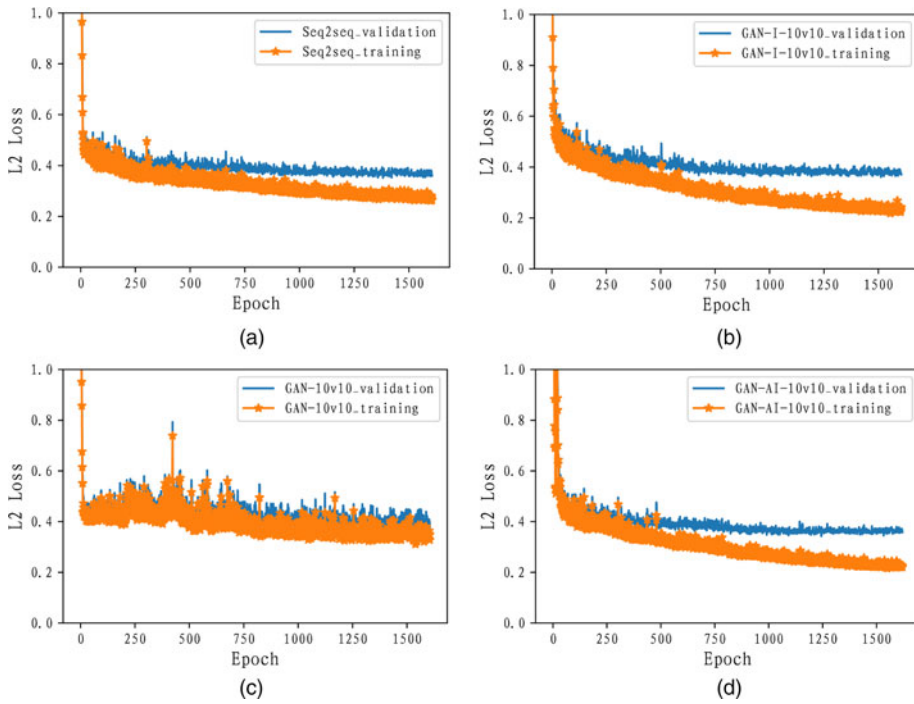
**4.1. Loss curves of training**

To evaluate performance of the GAN-AI, a series of models have been trained and named as ‘GAN-AI-KVN’, where ‘A’ denotes the GAN model assembles the attention module, and ‘I’ denotes the GAN model assembles the interaction module; ‘K’ means the model uses a variety loss function during the training; ‘N’ refers to the number of sampling time during test time and chooses the best prediction in L2 sense for quantitative evaluation. This paper compares the L2 loss-convergence curves of the classical trajectory prediction model sequence to sequence (Seq2seq), the plain GAN model GAN-10v10, and the improved GAN model: GAN-I-10v10 and GAN-AI-10v10. The initial learning rate is set to 0.001, and an Adam (Kingma and Ba, 2014) optimisation algorithm is used to optimise the network parameters. The batch size is set to 32, and the number of training iterations is set to 1,600. The observation sequence length is set to 9, and the prediction sequence length is set to 9. The sampling interval is 10 s.

The comparison results of the training curve and validation curve of the four models are shown in Figure 2. It can be seen from Figure 2 that the L2 loss of the four models shows a downward trend through iterative training and tends to be stable after 1,600 iterations. Among the four models, GAN-AI-10v10 has the best convergence, and the loss on the validation set is 0.331. The second best model is the GAN-I-10v10 model, with a loss of 0.358 on the validation set. The loss of the Seq2seq model on the validation data is 0.363. The GAN-10v10 model has the worst convergence, only 0.398. By observing the convergence trend of the convergence curve, it can be found that although GAN-10v10 finally converges, the convergence curve fluctuates greatly. But GAN-I-10v10 does not appear to have this phenomenon by assembling interactive modules, which can prove that interactive modules can effectively help the GAN network to achieve stable convergence. The GAN-AI-10v10 model achieves better convergence than does the GAN-I-10v10 model by assembling the attention module. It proves that the attention mechanism can help the network capture and grasp the motion information of ships at different times, thus improving the learning ability of the network.

**4.2. Accuracy and analysis**

To evaluate the trajectory prediction accuracy of the GAN-AI model, the average distance error (ADE) and final distance error (FDE) metrics used in reference Gupta et al. (2018) are introduced. ADE calculates the average Euclidean distance between the predicted trajectory and the real trajectory at each time point. The FDE calculates the average Euclidean distance between the predicted trajectory and the



**Figure 2.** Plots of models’ L2 loss curves: (a) is L2 loss curve of the Seq2seq; (b) is L2 loss curve of the GAN-I-10v10; (c) is L2 loss curve of the GAN-10v10; (d) is L2 loss curve of the GAN-AI-10v10;

real trajectory at the last time point of the prediction period. The formulae are:

$$ADE = \frac{\sum_{i=1}^n \sum_{t=t_{obs}+1}^{t_{pred}} \sqrt{((\hat{x}_i^t - x_i^t)^2 + (\hat{y}_i^t - y_i^t)^2)}}{n(t_{pred} - t_{obs})} \tag{27}$$

$$FDE = \frac{\sum_{i=1}^n \sqrt{((\hat{x}_i^{t_{pred}} - x_i^{t_{pred}})^2 + (\hat{y}_i^{t_{pred}} - y_i^{t_{pred}})^2)}}{n} \tag{28}$$

This paper has compared the prediction accuracy of six models: seq2seq, GAN-10v10, Kalman model, GAN-I-10v10, GAN-AI-10v10, and GAN-AI-20v20. Among them, seq2seq is the classical nonlinear model, the Kalman model is the classical linear model, and other GAN models are assembled different modules. Table 1 shows the comparison results of the trajectory prediction accuracy of six models on the test data set. It can be found that at the scene of ship encounter, by adding an interaction module, the model can learn ship interaction behaviour, effectively improve the prediction accuracy of the model, and make GAN-I-10v10, GAN-AI-10v10 and GAN-AI-20v20 obtain higher precision than other models without an interaction module. At the same time, by assembling the attention module, the model can improve the ability to extract effective data at multiple time points, so that the ADE of GAN-AI-10v10 is 1.8 m/3.0 m higher than that of GAN-I-10v10 at different prediction time length. Comparing the model accuracy of GAN-AI-10v10 and GAN-AI-20v20, it can be found that higher sampling times can effectively improve the prediction accuracy of the network during network training and evaluation, making the ADE of GAN-AI-20v20 model reach 28.4 m/40.7 m. In this paper, the Kalman prediction model gives a benchmark of the linear methods, which is compared with our non-linear method. . Because the linear methods can not model the non-linear motion of vessels and the ADE of the Kalman model only reaches 74.2 m/146.3 m at different prediction time lengths.

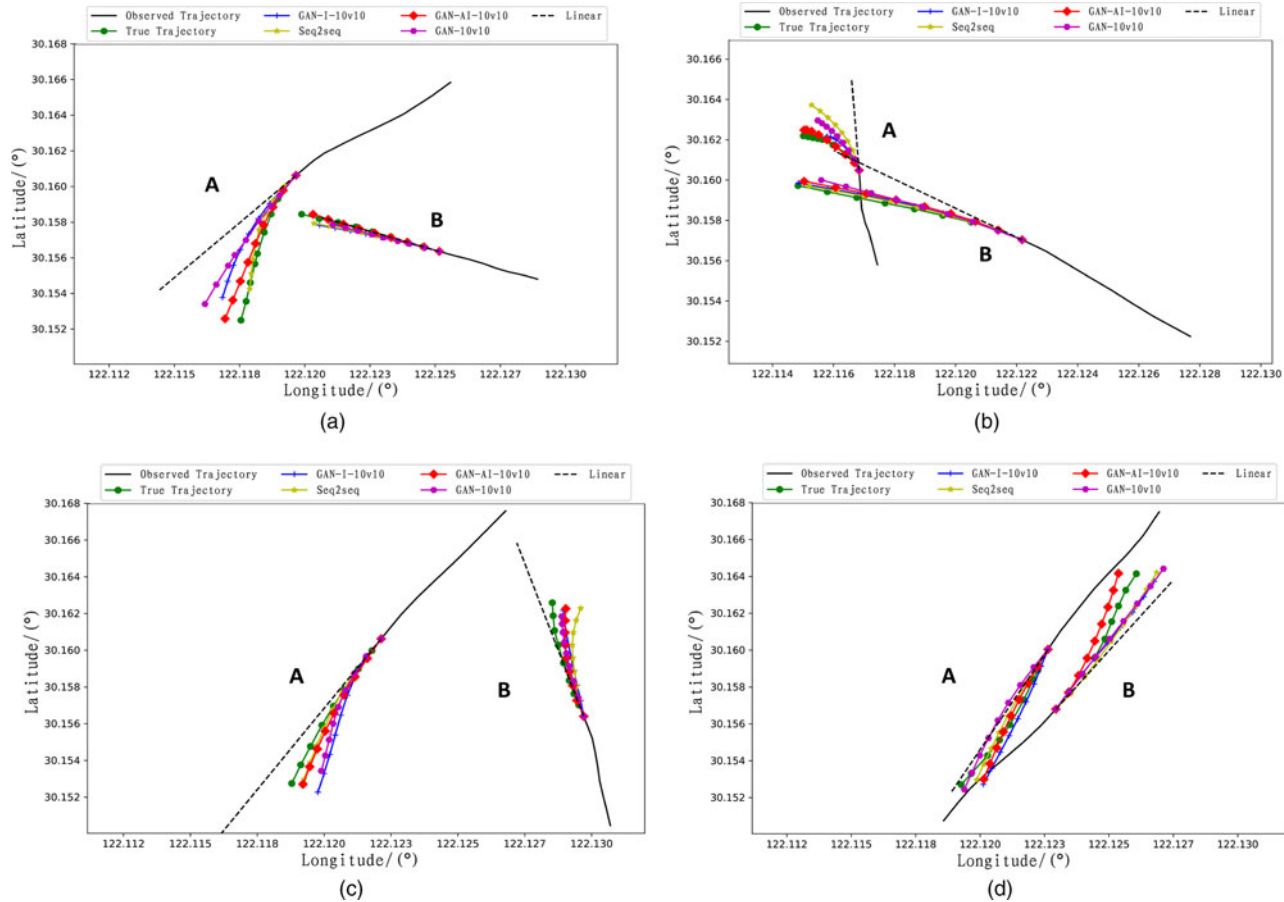
**Table 1.** Comparison of ADE and FDE of different models.

Prediction time length	Metrics	Seq2seq	GAN-10V10	GAN-I-10v10	GAN-AI-10v10	GAN-AI-20v20	Linear
90 s	ADE/m	36.1	37.6	34.7	32.9	28.4	74.2
	FDE/m	65.4	66.8	63.5	57.9	53.6	150.6
180 s	ADE/m	51.2	53.6	51.9	48.9	40.7	146.3
	FDE/m	98.7	103.3	97.8	93.8	82.4	284.8

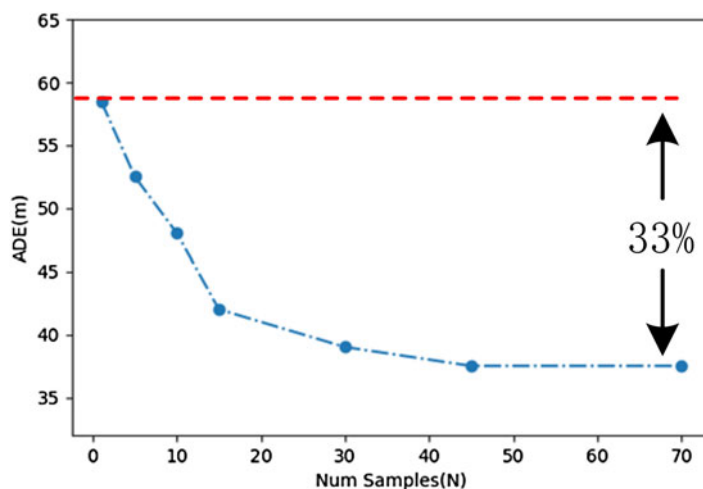
Although ADE and FDE can objectively evaluate the prediction accuracy of the model, it is unable to analyse the reasons for the differences in prediction accuracy of different models. Figure 3 compares the prediction differences of five models in four typical real encounter scenarios. Figure 3a is the prediction result of two ships in the crossing situation, Ship A is an accelerating ship to enter the port after judging the situation, and Ship B keeps straight-line navigation. Comparing the predicted results about Ship B, it can be found that the predicted results of the five models are similarly consistent, and they are consistent with the distribution of the real trajectory in the navigation distance and direction. However, comparing the prediction results about Ship A, Kalman model’s ADE/FDE only reaches 197.3m/409.1m that achieved the largest prediction error, as it is a linear model that cannot effectively estimate the turning behavior of the ship. Comparing to the linear method, the other four nonlinear methods can better estimate the navigation trend of the ship’s future trajectory. But comparing to the GAN-AI-10v10, the GAN-I-10v10, seq2seq and GAN-10v10 have greater errors in the navigation distance and course. Figure 3b shows the prediction results of two ships in the real overtaking situation. Ship A is a slowly left-turning ship. After judging the situation, Ship B takes the action of overtaking from Ship B’s port side. Comparing the prediction results about Ship A, the prediction results of GAN-AI-10v10 are consistent with the real trajectory, and ADE/FDE is 24.5 m/25.7 m. The GAN-10v10 and GAN-I-10v10 models have errors in the estimation of the navigation speed, but the course estimation is consistent with the real result, while the Seq2seq model is consistent with the real result in the speed, but the course estimation has errors. Comparing the prediction results about Ship B, excepting for the linear method, the other four models are consistent with the real trajectory. Figure 3c is the prediction result in the free situation. Although Ships A and B are in the same water area, they do not have interactive behaviour. For the five prediction results, excepting for the distortion of the linear prediction method, the prediction results of the other four models are similar to the real trajectory temporal–spatial distribution. Figure 3d is the prediction result of the head-on situation. Ship A drives in a straight line after the encounter and turns right at the later part of the journey. Ship B turns left and adjusts to the original route after the encounter. Comparing the prediction results about Ship A, the prediction results of GAN-AI-10v10 and seq2seq models are better in the former section, but the prediction of the turning track in the latter section has deviation, and the ADE/FDE are 47.5 m/95.8 m and 49.6 m/101.4 m. Among the five models, GAN-10v10 is the best, ADE/FDE is 46.6 m/33.6 m. The results of GAN-I-10v10 were the worst, and ADE/FDE is 77.6 m/178.4 m. Comparing the results of ship B, the error of GAN-AI-10v10 and GAN-I-10v10 is smaller, and the ADE/FDE are 45.8 m/77.4 m and 47.1 m/81.3 m. The results of seq2seq and gan-10v10 are similar to the linear method, and ADE/FDE are greater than 50 m/85 m.

By comparing the trajectory prediction results in four different scenarios, it can be found that the reason for GAN-AI-10v10 attains higher accuracy is that it can better estimate the parameters of course and distance simultaneously in four scenarios. By assembling the interactive module, the GAN-AI-10v10 model can better learn the ship’s course and speed changes. By assembling the attention module, the GAN-AI-10v10 model can better allocate the parameter weight of the model and improve the spatiotemporal modelling ability of the model. On the other hand, the GAN-I-10v10 model has a wrong estimation of speed in the case of Figures 3a and 3b due to the lack of attention mechanism. The seq2seq and GAN-10v10 models have error estimation of course in Figures 3a and 3b due to the lack





**Figure 3.** Prediction results of four different situations: (a) is the result in the crossing situation; (b) is the result in the overtaking situation; (c) is the result in the free situation; (d) is the result in the head-on situation.



**Figure 4.** Effect of variety loss.

of interaction modules. It indicates that the attention module and interaction module can improve the accuracy of speed and course estimation.

### 4.3. Analysis of variety loss

A series of models of GAN-AI-NVN have been trained to evaluate the influence between the performance and the variety loss function. ‘NvN’ indicates the GAN-AI using the variety loss function during the training. ‘N’ refers to the number of times we sample from GAN-AI model during test time. As shown in Figure 4 the model can get better performance with the increase of  $N$  and the performance gets about 30% better with  $N = 70$ . On the other hand, the model’s computation will be increasing accordingly with  $N$  and the rise of accuracy will be slower. As a result,  $N = 20$  is a pretty setting for computation and accuracy.

## 5. Conclusion

A GAN-AI model is proposed to predict the trajectory of multiple ships in the same water area simultaneously. In the GAN-AI model, an attention module is designed to improve the ability of the network to extract effective data and avoid memory degradation of LSTM. An interaction module is designed to model the interaction behaviour information, such as relative distance and relative speed between ships, so as to get better performance in complex interactive scenes. To improve the convergence ability and prediction accuracy of the network, the adversarial loss function and variety loss function are introduced into the network training. Using the ADE and FDE metrics to verify the GAN-AI model on the real AIS historical data, the results show that the ADE/FDE of the GAN-AI-20V20 model is less than 40.7 m/82.4 m under the prediction duration of 180s. Compared with seq2seq, plain GAN and Kalman models, the performance has been improved by 20%, 24% and 72%, respectively. It is of great significance to improve the strength of the vessel traffic management system, the path planning of intelligent ships and the analysis of ship collision risk. On the other hand, more AIS data should be extracted to validate the model’s generalisation ability and data fusion can also be utilised to improve the accuracy further in future work.

**Funding statement.** The research reported here was sponsored by Sanya Science and Education Innovation Park of Wuhan University of Technology (Grant No. 2020KF0040), the Key Research Plan of Zhejiang Provincial Department of Science and Technology (Grant No. 2021C01010).

**Competing interests.** The authors declare no conflicts of interest.

## References

- Bahdanau, D., Cho, K. and Bengio, Y.** (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
- Borkowski, P.** (2010). Computational mathematics in marine navigation. *Zeszyty Naukowe/Akademia Morska w Szczecinie*, **21**, 20–27.
- Borkowski, P.** (2017). The ship movement trajectory prediction algorithm using navigational data fusion. *Sensors*, **17**, 1432.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.** (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, **3**, 2672–2680.
- Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S. and Alahi, A.** (2018). Social gan: Socially Acceptable Trajectories with Generative Adversarial Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2255–2264.
- Huang, H.** (2018). Trajectory Prediction for Ocean Vessels Base on K-Order Multivariate Markov Chain. *Proceedings of the Wireless Algorithms, Systems, and Applications: 13th International Conference, WASA 2018, Tianjin, China, June 20–22, 2018*, 140.
- Kaluza, P., Kölzsch, A., Gastner, M. T. and Blasius, B.** (2010). The complex network of global cargo ship movements. *Journal of the Royal Society Interface*, **7**, 1093–1103.
- Kingma, D. P. and Ba, J.** (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Krzysztof, J.** (2017). Automatic identification system (AIS) dynamic data estimation based on discrete Kalman Filter (KF) algorithm. *Scientific Journal of Polish Naval Academy*, **211**, 71–87.
- Li, W., Zhang, C., Ma, J. and Jia, C.** (2019). Long-term Vessel Motion Prediction by Modeling Trajectory Patterns with AIS Data. *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, 1389–1394.
- Liu, C., Guo, S., Feng, Y., Hong, F., Huang, H. and Guo, Z.** (2019a). L-VTP: Long-term vessel trajectory prediction based on multi-source data analysis. *Sensors*, **19**, 4365.
- Liu, J., Shi, G. and Zhu, K.** (2019b). Vessel trajectory prediction model based on AIS sensor data and adaptive chaos differential evolution support vector regression (ACDE-SVR). *Applied Sciences*, **9**, 2983.
- Luong, M.-T., Pham, H. and Manning, C. D.** (2015). Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025.
- Perera, L. P. and Soares, C. G.** (2010). Ocean Vessel Trajectory Estimation and Prediction Based on Extended Kalman Filter. *The Second International Conference on Adaptive and Self-Adaptive Systems and Applications*.
- Sainath, T. N., Vinyals, O., Senior, A. and Sak, H. C.** (2015). Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 4580–4584.
- Sun, L. and Zhou, W.** (2017). Vessel Motion Statistical Learning Based on Stored AIS Data and Its Application to Trajectory Prediction. *2017 5th International Conference on Machinery, Materials and Computing Technology (ICMMCT 2017)*. Atlantis Press, 1183–1189.
- Suo, Y., Chen, W., Claramunt, C. and Yang, S.** (2020). A ship trajectory prediction framework based on a recurrent neural network. *Sensors*, **20**, 5133.
- Sutulo, S., Moreira, L. and Soares, C. G.** (2002). Mathematical models for ship path prediction in manoeuvring simulation systems. *Ocean Engineering*, **29**, 1–19.
- Tang, H., Yin, Y. and Shen, H.** (2019). A model for vessel trajectory prediction based on long short-term memory neural network. *Journal of Marine Engineering & Technology*, 1–10. doi: 10.1080/20464177.2019.1665258
- Tu, E., Zhang, G., Rachmawati, L., Rajabally, E. and Huang, G.-B.** (2017). Exploiting AIS data for intelligent maritime navigation: A comprehensive survey from data to methodology. *IEEE Transactions on Intelligent Transportation Systems*, **19**, 1559–1582.
- Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y. and Change Loy, C.** (2018). Esrgan: Enhanced Super-Resolution Generative Adversarial Networks. *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Zhang, X., Liu, G., Hu, C. and Ma, X.** (2019). Wavelet Analysis Based Hidden Markov Model for Large Ship Trajectory Prediction. *2019 Chinese Control Conference (CCC)*, 2913–2918.