

# Research On Cruise Trajectory Prediction During Voyage Test Based On AIS Data

Hong Zhou, Qiyi Zhang

**Abstract**— In the course of voyage test, cruise ships usually chose preset voyage forward, but sometimes captain would choose to operate the ship by himself. Based on the analysis of cruise ship accidents over the years, it was found that the accident of the cruise ship hitting rocks or colliding occurs frequently due to the captain's subjective preference in choosing the route. The purpose of this paper was using python to make a timely judgment of the risk when the captain deviated significantly from the scheduled route, but it was not necessarily dangerous to stay away from the scheduled route, and the reliable basis to ensure the safety of the route was that there have been cruises that have chosen to sail in parallel. This paper helped find a way to judge whether the route is dangerous or not. The basis for judging whether the route dangerous or not is the large gap between the current route and the predicted route after the classification and regression of the cruise's. Therefore, this article need to find a way to simply and precisely forecast steps on the captain's route choice for dangerous discernment.

**Index Terms**— voyage test; cruise; trajectory prediction; AIS data

## I. INTRODUCTION OF CRUISE TRAJECTORY PREDICTION

Cruise track prediction was to judge the trajectory that cruise will choose at the next moment or in the next period of time according to the historical track of cruise and classified and regressed data, so as to warn or remind the operator of the risk of the current path when the cruise ship deviated significantly from the preset track.

Cruise line prediction was based on a large number of tracks, which needed to be screened, classified and predicted by corresponding software. As a free and open source software, python could not only make use of algorithms to train dynamic data of multiple types of cruises, so as to form a model with high accuracy, but also be portable and extensible. This chapter would use python, cooperating with several common and suitable algorithms: KNN, SVM, RNN-LSTM algorithm and Naive Bayesian algorithm, to analyze the accuracy of cruise trajectory prediction and the advantages and disadvantages of each algorithm, so as to study whether trajectory prediction with python was feasible and which algorithm was the most appropriate for trajectory prediction.

The cruise ship studied in this chapter was a new cruise ship built by SHANGHAI WAIGAOQIAO SHIPBUILDING Co (S.W.S.C). The starting position of the cruise ship was SWSC. When choosing the cruise track, the most valuable track was

its nearest arrival at Shanghai Wusongkou International Cruise Terminal and the track of the cruise ship departing from this port.

## II. DEFINITION AND EXTRACTION OF AIS DATA

Cruise AIS data collected mainly include static data, dynamic data and voyage data, sample of AIS data shown in table I . The static data consists of cruise identification information and cruise scale, which include ship name, call number, MMSI, ship type, length of design waterline, molded breath and so on. MMSI is ship radio communication identification code; Cruise dynamic data refers to the operating route of cruise ship during navigation, which mainly includes positioning information, track, navigational speed and track Angle. Cruise voyage data describes cruise voyage status, including draft depth, port of passage and destination. One of the main research purposes of this paper is to predict the cruise ship's voyage test track. Therefore, the collected AIS data are derived from voyage process and the voyage status of AIS data is voyage.

At present, researchers can extract AIS data from a variety of perspectives. When the navigation state of the cruise ship changes from anchoring to anchoring, this point is taken as the cut-off point of AIS track data. This data extraction method generally sets the AIS data of cruise ship as  $T=\{p_1,p_2,p_3,\dots,p_n\}$ . Then, the steps of extraction of route trajectory data sequence based on the navigation state are as followed:

Step 1: initialize the ship track sequence set S;

Step 2: traverse the data sequence T until the data point  $P_j$  is found. The current sailing state of  $P_j$  is "anchoring" or "berthing".

Step 3: start from  $P_j$  and traverse until find  $p_k$ . The current sailing state of  $P_k$  is "under sail". Then, create new trajectory sequence  $s=\{P_j,P_{j+1},\dots,P_{k-1}\}$  and add s to S after that repeat step 2. At last, start with k and repeat step 2 until all data are traversed.

After completing above steps, track set S is the track sequence set of all cruise ships. The research object of this chapter was newly built cruise ships. Therefore, when choosing route, the port was not the load-point. After observing several cruise line nearby the port, the paper found that the route become more and more unstable when the cruise approach to port, shown in Fig 1 . Routes like this would make the prediction become incorrect Thus, the following research chose to neglect the route closed to port.

**Hong Zhou**, Professor,, Jiangsu university of science and technology, China

**Qiyi Zhang**, Graduate student, Jiangsu university of science and technology, China

Table I Sample of cruise AIS data

ID	MMSI	Longitude	Latitude	Speed	Angle	Time
2019/07/25						
1110	311000396	63°3.433	18°0.522	5.2	79	19:05:01
1111	311000396	63°3.198	18°0.564	5.2	108	19:07:41
1112	311000396	63°3.126	18°0.541	3.2	162	19:08:44
1113	311000396	63°3.104	18°0.481	2.5	261	19:10:21
1114	311000396	63°3.142	18°0.473	1.0	49	19:15:52
1115	311000396	63°3.08	18°0.525	2.0	48	19:18:50
1116	311000396	63°2.954	18°0.631	1.2	33	19:24:10
1117	311000396	63°2.917	18°0.685	0	0	19:39:06
2019/07/26						
1118	311000396	63°3.041	18°0.56	4.2	239	05:02:35
1119	311000396	63°3.134	18°0.506	4.9	257	05:03:56
1120	311000396	63°4.047	18°0.304	12.71	260	05:09:55
1121	311000396	63°5.319	18°0.102	16.11	260	05:14:58

As shown in the Fig I , the route of cruise got complex when cruise became more and more close to the port. In order to get useful route data, it's important to find a proper point to cut the route into the most valuable pieces. Therefore, the paper chose data, whose SOG was above a certain velocity. The paper collected several data from AIS data close to the port, as shown in Fig II and Fig III. The paper found that the SOG of the cruise became chaos along with the COG.

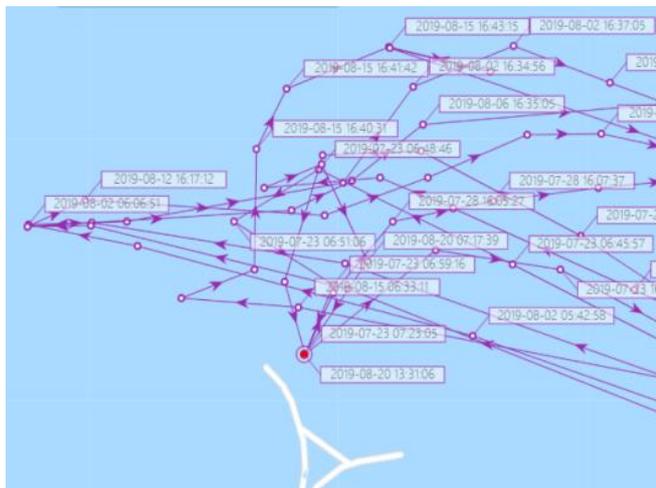


Fig I . AIS data map nearby the port

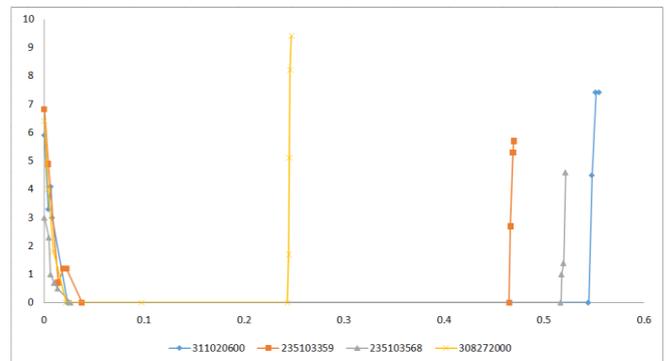


Fig III. Scatter diagram of cruise liner track with stable change of track Angle and departure velocity

The route passed Shanghai Wusongkou International Cruise Terminal must go through the channel beside the SWSC. For the sake of making AIS data as useful as possible, the paper chose a certain speed, such as 7kn. Before extracting the trajectory sequence based on navigation, set the cruise AIS data as data sequence  $T = \{p_1, p_2, p_3, \dots, p_n\}$ . Then, the algorithm of selecting the AIS data, whose SOG was above 7kn, is followed.

- Step1: initializing the cruise trajectory sequence collection S.
- Step2: traverse the data sequence T until finding data point  $P_j$ , whose SOG is 7kn. Then, set the point as the cut point.
- Step3: set the point as the starting point and repeat the step2, until all the data got traversed.

After finishing the above steps, the paper got several cruise trajectory sequences. The trajectory sequences samples were followed:

Table II. Route data

ID	POINT
$S_1$	$(P_{11}, P_{12}, P_{13}, \dots)$
$S_2$	$(P_{21}, P_{22}, P_{23}, \dots)$
$S_3$	$(P_{31}, P_{32}, P_{33}, \dots)$
$S_4$	$(P_{41}, P_{42}, P_{43}, \dots)$
$S_5$	$(P_{51}, P_{52}, P_{53}, \dots)$
$S_6$	$(P_{61}, P_{62}, P_{63}, \dots)$
$S_7$	$(P_{71}, P_{72}, P_{73}, \dots)$
$S_8$	$(P_{81}, P_{82}, P_{83}, \dots)$

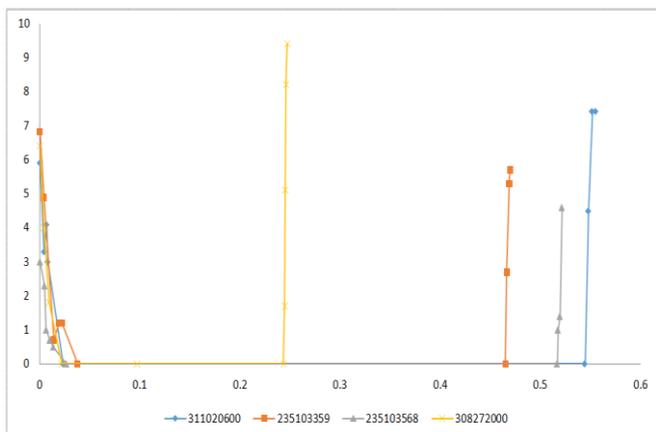


Fig II . Scatter diagram of velocity variation of cruise liner track with unstable change of track angle near the port

III. CRUISE TRAJECTORY PREDICTION ALGORITHM BASED ON AIS DATA

The mathematics description of trajectory prediction algorithm is setting the trajectory data of selected cruise as  $T=\{t_1,t_2,t_3,\dots,t_n\}$ , and  $t_i$  means the NO.i trajectory point. The description set trajectory collection as  $S=\{s_1,s_2,s_3,\dots,s_m\}$  and  $s_i$  means the NO.i trajectory piece. Afterwards, the aim of the algorithm is to find a mapping function (1), which could make function (2) become as small as possible.

$$f(\bullet): \hat{Y}_i = f(t, s) \quad (1)$$

$$\text{Min}(|\hat{Y} - Y|) \quad (2)$$

Where  $Y$  and  $\hat{Y}$  were the real and predicted route of cruise.  $(\hat{Y} - Y)$  was the difference between real route and predicted route. While the difference was smaller, the more accurate the prediction would be. The mapping function could be assumed as the most similar and possible route extracted from history routes.

Before forecasting the route by algorithm, the sample set should be divided into training set and testing set. General evaluating way covers "hold-out" and "cross validation". The "hold-out" was separating the data set into two mutually exclusive sets. However, the "cross-validation" divided the data set  $D$  into  $k$  mutually exclusive subsets of similar size, which means  $D=D_1 \cup D_2 \cup \dots \cup D_k$ ,  $D_i \cap D_j = \emptyset$  ( $i \neq j$ ). Generally, the "hold-out" was easy to separate and common. However, the smaller the sample set was, the more difficult the separating way would be. Because, as long as the increase of training set proportion, the differences between training set and testing set added. Then, the differences between evaluating model and training model added. Nevertheless, as long as the decrease of training set proportion, the training model was more and more closed to the training set, which made the evaluating model become inaccurate. This paper would separate the sample set into 8:2. The training set would occupy the 80% of the sample and the testing set would occupy the 20% of the sample set.

A. Cruise trajectory algorithm based on SVM-KNN

The nearest neighbor algorithm was a common classification algorithm, whose basic idea was to find the individual with the least difference from the predicted sample, and thought that the category of the predicted sample was consistent with the category of the individual. The differences between samples could be measured by space length and similarity. When the space length was getting huge, the difference was getting larger. When the similarity was getting huge, the difference was getting larger[1].

If the nearest neighbor algorithm only used the individual closest to the predicted sample as the observation sample, large errors would be generated and overfitting will easily occur. In particular, when the observed individual was noisy data, it had a great influence on the prediction. Therefore, the optimization method of the nearest neighbor algorithm was  $K$  nearest neighbor (KNN), and better prediction results could be obtained by observing multiple individuals with small differences from the predicted samples. The basic idea of  $k$ -nearest neighbor was to find the  $K$  samples closest to the prediction sample, which were called neighbors, and then

choose the category with the highest votes by voting. In general, the difference between the current sample and  $K$  neighbors was different. In category discrimination, each neighbor had different influence on the result. Thus, the weight of each neighbor was different. Therefore, the weighted  $k$ -nearest neighbor algorithm was derived.

$D$  was used to represent the difference measurement between  $K$  nearest neighbors and the target individual, and  $W$  was used to represent the influence weight of  $K$  nearest neighbors on the target individual.

$$\text{For } \sum W=1, \text{ if } D_i < D_j, W_i > W_j \quad (3)$$

It means that the smaller the differences were, the bigger the weight of influence would be. When using KNN, make sure the weight vector is key point to turn  $W$  into a function of  $D$  (4). Then normalize  $W$ .

$$W = f(D) \quad (4)$$

Another way was setting fixed step value for  $W$  and  $C$  was used to represent the value set of category.  $R$  represented the  $K$  nearest categories of target unit.

$$\text{If } R_i = C_j, R_i C_i = 1, \text{ else } R_i C_i = 0 \quad (5)$$

The probability of category  $C_j$  is :

$$P(C_j) = \sum R_i C_i \cdot W_i \quad (6)$$

The individual category discriminant is:

$$\text{argmax} \sum R_i C_i \cdot W_i \quad (7)$$

The basic logic of support vector machine (SVM) is to find a partition hyperplane applied to divided the sample  $D = \{(x_1, y_1); (x_2, y_2); \dots; (x_m, y_m)\}$ ,  $y_i \in \{-1, +1\}$  into different types. In general, the choice of hyperplane tried to ensure that the results generated after hyperplane partition were the most robust and the generalization ability for the unseen examples was the strongest. In the sample space, the hyperplane partition could be described by the following linear equation[2]:

$$\omega^T x + b = 0 \quad (8)$$

The  $\omega = \{\omega_1; \omega_2; \dots; \omega_d\}$  was a normal vector, which determines the direction of the hyperplane;  $B$  was the displacement term, which determined the distance between the hyperplane and the far point. Obviously, the partition hyperplane could be determined by the normal vector  $\omega$  and displacement  $b$ , which we would call  $\omega b$ . In the sample space, the distance from any point  $x$  to the hyperplane could be written as:

$$r = \frac{|\omega^T x + b|}{\|\omega\|} \quad (9)$$

Assumed that the hyperplane sample could correctly classify the training sample, for  $(x_i, y_i) \in D$ , if  $y_i = +1$ , then  $\omega^T x_i + b > 0$ ; if  $y_i = -1$ , then  $\omega^T x_i + b < 0$ . Support vector could make followed equation of the training samples closest to the hyperplane come true.

$$\begin{cases} \omega^T x_i + b \geq +1, & y_i = +1 \\ \omega^T x_i + b \leq -1, & y_i = -1 \end{cases} \quad (10)$$

The sum of the distance from the two heterogeneous support vectors to the hyperplane was called interval. The equation and figure were followed.

$$r = \frac{2}{\|\omega\|} \quad (11)$$

When the constraint parameter normal vector  $\omega$  and displacement  $b$  were found, the gamma could be maximized. Then, the partition hyperplane with the maximum constraint interval could be found. The equation were followed.

$$\max_{\omega, b} = \frac{2}{\|\omega\|} \quad (12)$$

$$\text{s.t. } y_i(\omega T x_i + b) \geq 1, \quad i=1,2,3,\dots,m \quad (13)$$

Obviously, in order to maximize the interval, just maximized  $\frac{1}{\|\omega\|}$ , which was the same thing as minimizing  $\|\omega\|^2$ . The function could be rewritten as:

$$\max_{\omega, b} \frac{1}{2} \|\omega\|^2 \quad (14)$$

$$\text{s.t. } y_i(\omega T 14 x_i + b) \geq 1, \quad i=1,2,3,\dots,m \quad (15)$$

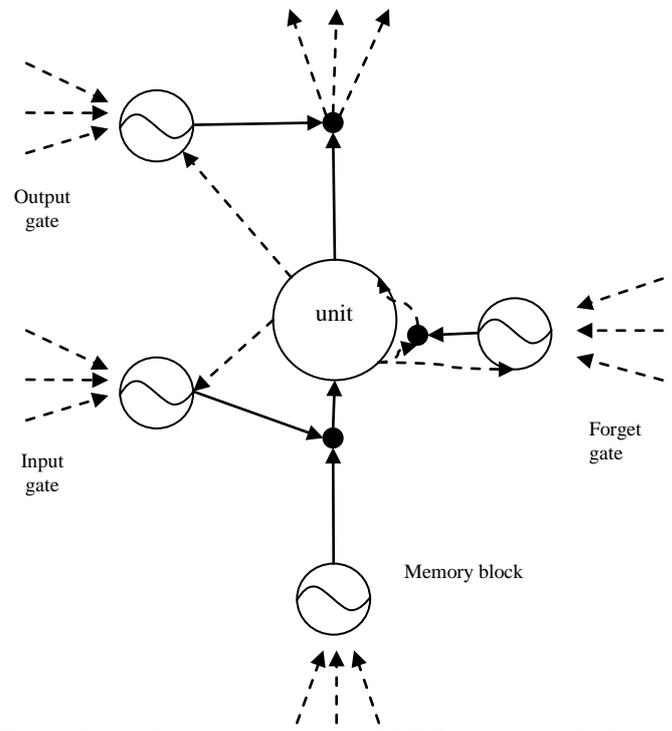
$\xi$  was the basic support vector machine. At the same time, in order to reduce the errors near the hyperplane SVM algorithm classification, when choosing the threshold factor,  $\xi$  needed more than  $\frac{1}{\|\omega\|}$ .

### B. The principle of RNN-LSTM algorithm

LSTM model was a special RNN model, which was mainly used to solve the problem of gradient disappearance in ordinary RNN model when the time span of input samples was long[3]. Compared with the general neural network, RNN model had an irreplaceable advantage -- memory function. At the same time, like the basic neural network model, it could combine the low-level features by learning to form the high-level features, and then discovered the distributed features of the training set. It was proved that LSTM had excellent performance when dealing with many modeling problems. LSTM could not only be used to learn the shallow nonlinear network structure, approximating complex functions, and finally obtained the essential characteristics of the input sample data. More importantly, it solved the dilemma that the low information utilization rate of the standard RNN long-term samples led to the disappearance of the gradient, and ensured the realization of long-term information memory.

LSTM model was established on the basis of TIME series data, while AIS data included TIME, which could be changed as the INTERVAL of characteristic data in LSTM. The trajectory characteristic  $Y(t)$  of the cruise ship at time  $t$  that could be written as  $Y(t) = \{itv, lng, lad, sog, cog\}$ . Cruise trajectory characterization data  $Y(t+1)$  was essentially the prediction result of the first  $n$  moments, which could also be called output gate. The characteristic data of ship trajectory at the first  $n$  moments  $Y(t-n+1), Y(t-n+2), Y(t-n+3), \dots, Y(t)$  was the input gate, which was shown in fig5.  $Y(t+1)$  could be illustrated as function 19.

$$Y(t+1) = f(\{Y(t-n+1), Y(t-n+2), \dots, Y(t)\}) \quad (19)$$



FigIV. Internal structure diagram of LSTM memory block

In LSTM training, the prediction accuracy of neural network was affected by the number of LSTM layer neurons (memory modules). In the training, the empirical formula was used as the initial value to determine the valuation, and the appropriate number of neurons was calculated through experiments. At last the number with the minimum error is identified as the number of LSTM layer nodes in the training.

### C. The principle of Naive Bayesian(NB) algorithm

The first thing to think about when using NB models was bayesian decision theory. In theory, bayesian decision theory was a theory that could select the optimal category after comparing the correlation probability and the loss caused by misclassification of the sample set in a sample set when all relevant probabilities were known [4].

Cruise ship's voyage test path was generally divided into two parts, the open sea and the offshore. In the offshore part, the deviation between cruise lines in the sea trial was relatively small. However, in the offshore part, especially the route near the port and the route away from the port, there were more cases that the captain operates the route, and many serious cruise accidents occurred near the cruise port or the area close to the coastline. This phenomenon was generally attributed to the following reasons, such as the captain's negligence or deliberate searching for a sense of accomplishment, the lack of personnel management in the process of operation, etc. In the far-sea part, there were preset routes, and the preset routes were mostly the routes with high frequency, so there were few environmental risk factors. However, NB generally only needed the correct ranking of conditional probability of various categories, and could correctly classify the results on the basis of no accurate probability value, which was suitable for the data of the thin and long-type region such as the navigation track.

BN assumed that training set had N different types[5], then , the training set was  $Y=\{c_1 \ 、 c_2 \ 、 c_3 \ 、 \dots \ 、 c_N\}$  . Afterwards, BN assumed  $\lambda_{ij}$  was the loss due to misclassification of samples belonging to  $c_j$ . The expected loss caused by misclassifying sample  $x$  into  $c_i$  could be determined by the posterior probability  $P(c_i|x)$  , namely  $R(c_i|x)$  , which was the conditional risk of sample  $x$ . The task of this section was to find the minimum overall risk.

$$R(c_i|x) = \sum_{j=1}^N \lambda_{ij} P(c_j|x) \quad (20)$$

$$R(x) = E_x[R(h(x|x))] \quad (21)$$

Therefore, if wanted to minimize  $R(h)$ , just minimized the conditional risk  $R(h(x|x))$ . After that, it was necessary to select the category marker that could minimize the conditional risk on each sample based on the bayesian criteria. That is:

$$h^*(x) = \arg \min_{c \in Y} R(c|x) \quad (22)$$

In the above equation,  $h^*$  was called the optimal classifier, and the corresponding overall risk  $R(h^*)$  was called bayesian risk, and  $1-r(h^*)$  was the optimal performance that the classifier could achieve in this training set. When the goal was to minimize the classification error rate, the misjudgment loss could be written as:

$$\text{If } i=j, \text{ then } \lambda_{ij}=0, \text{ otherwise } \lambda_{ij}=1 \quad (23)$$

The conditional risk in this case is:

$$R(c|x) = 1 - P(c|x) \quad (24)$$

Thus, the optimal bayesian optimal classifier, which minimized the classification error rate, could be obtained:

$$h^*(x) = \arg \min_{c \in Y} P(c|x) \quad (25)$$

The function helped select the category marker that maximized the posterior probability  $P(c|x)$  .In the actual training process, the a posteriori probability  $P(c|x)$  was hard to get. Only in the process of machine learning through a training on the training sample set, that was able to make the a posteriori probability as close as possible to the real  $P(c|x)$  .However, to obtain  $P(c|x)$  as accurately as possible, there were two approaches. One approach was discriminant model, in which,  $x$  had been given and  $c$  was predicted by modeling  $P(c|x)$  . The SVM mentioned above was discriminant model.The other was the joint probability distribution  $P(c|x)$  modeling, so as to obtain  $P(c|x)$  again, namely the generative model. Before generating the model, the following function (26) must be satisfied.

$$P(c|x) = \frac{P(x, c)}{p(x)} \quad (26)$$

Based on Bayes theorem, the formula  $P(x|c)$  could be written as function (27):

$$P(c|x) = \frac{P(c)P(x|c)}{p(x)} \quad (27)$$

In the above formula,  $P(c)$  was a kind of prior probability, while  $P(x|c)$  was a kind of conditional probability of sample  $x$  relative to category marker  $c$ , which could also be called

"likelihood".  $P(x)$  was the "evidence" factor used for normalization. When sample  $x$  was given, the evidence factor  $P(x)$  was not related to the category marker, and then the focus of the problem was how to estimate prior  $P(x)$  and likelihood  $P(x|c)$  on the basis of training set  $D$ .

#### IV. EXPERIMENTAL ANALYSIS

The data selected in this paper were from AIS data service platform. When dividing the test set, the data set was divided into training set and test set according to the ratio of 8:2, among which there were 4985 training set data and 1336 test set data. The research divided different trajectory segments according to 3 hours and put randomly selected data of trajectory segments into training and test sets., as shown in table III.

Table III. The table of the selected training and test sample

number	training	testing	proportion	times
6231	4985	1336	8:2	50

##### A. The Validation of data set

Route prediction algorithm was used to conduct training in the above training set and model performance evaluation on the test set. The algorithm was divided into two tasks:

- 1.Route classification task to predict which type of route the AIS data belongs to.
- 2.Predict the position information of AIS data in the future.

In order to verify the prediction effect of each algorithm in the course trajectory prediction, this paper carried out experiments in the data described in the previous section. Due to the annotated data in this data set, the route prediction problem was defined as a classification problem, and the classification accuracy was used to evaluate the experimental results. Accuracy was defined as follows:

$$\text{Prediction} = t/(t+q) \quad (28)$$

In above formula,  $t$  represented the number of tracks predicted correctly in the ship track, and  $q$  represented the number of tracks predicted incorrectly in the ship track

The accuracy of track prediction is defined in equations (29) and (30).

$$\text{RMSE} = \text{dist}(o_i, p_i) \quad (29)$$

$$\text{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (o_i - p_i)^2} \quad (30)$$

In above formula,  $p_i$  represented the predicted route,  $o_i$  represented the real route, and  $\text{dist}$  was the distance function. When calculating the distance between two routes, the measurement method of track similarity mentioned in this paper was used to calculate. If  $\text{RMSE} \geq d$ , the prediction was wrong. If  $\text{RMSE} < d$ , the prediction was right.

##### B. Settings of algorithm parameter

The route prediction algorithm was used to conduct experiments in the above two data sets, and the global parameters of each algorithm were set as followed: based on the regional statistics algorithm; area size width=2, height=2; Weighted K nearest neighbor: the weighted function was  $i/2$ , and the weighted function used  $i/2$ . The letter represents the  $i$ th nearest neighbor, and the number of nearest neighbors was  $K=\{3,5\}$ .

In this paper, LSTM adopted the strategy of ten-fold-cross validation. A set of data based on time series of the same ship was normalized and input as the model of a single target. In neural network, only the track feature vectors of multiple input moments were splicing, and the data format was long vector. However, in the input of LSTM network, since LSTM was based on time series, the input data format was a second-order matrix. As an important parameter of the experiment, the number of specific time points of the time series also affected the number of neurons in the LSTM layer of the trajectory prediction model. Due to the need to directly compare the prediction effects of the two prediction models, in order to be simple, the evaluation index only adopted the MSE mean square error method mentioned in 4.1.1 to evaluate the ship track prediction model. Mean square error referred to the expected value of the squared difference between observed value of parameter observed and predicted of parameter true value. The smaller the value of MSE was, the higher the accuracy of the experimental data described by the trajectory prediction model was.

C. Experimental results and analysis

Fig V was the selected sailing track, and T1, T2 and T3 were the selected predicted positions respectively. The analysis of experimental results could be divided into two parts:

1. Prediction effect of airline classification model using traditional machine learning algorithm and performance comparison.
2. Track regression prediction model using traditional machine learning and deep learning algorithms.

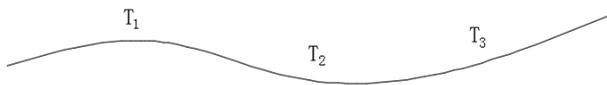


Fig V. The trajectory curve of Sea Of The Spectrum used for prediction

For the route classification model, NB and SVM algorithm model were used in the experiment to predict the route from the input AIS data. A variety of prediction methods were used to conduct experiments on the data set, and the experimental statistical results were shown in table 4, where the horizontal axis represented the AIS data input that had traveled the route, and the vertical axis represented the prediction accuracy, expressed in percentage. From the overall trend of figure7, the trend of each curve was gradually increasing, indicating that the accuracy of prediction from T1 to T3 was constantly improving. This was because that the ship track from T1 to T3 was known to be longer and longer, and there were more and more descriptions of ship track, so the prediction was more and more accurate. Look from the prediction results, finding that the best was the SVM prediction effect model effect that was significantly better than simple bayesian model. What's

more, the predictive accuracy of predicted points in T3 was more than 90%. This was mainly because less data used in this article AIS and the fitting degree of the SVM algorithm for a small amount of sample were better, helping achieve good results in classification and prediction. SVM-KNN had better performance, but more complex in implementation.

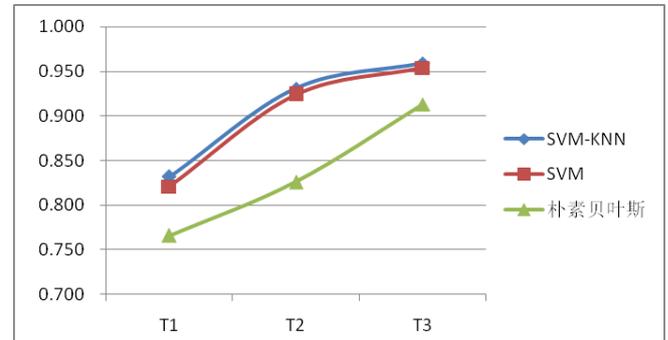


Fig VI. SVM-KNN SVM and NB accuracy comparison

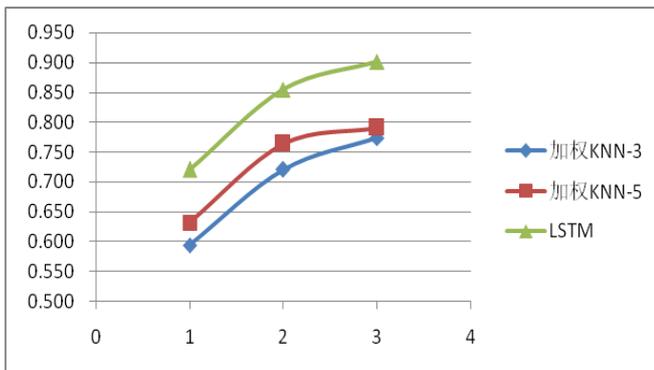
Table IV. The table of SVM-KNN SVM and NB accuracy comparison

Prediction algorithm	Prediction point		
	T1	T2	T3
SVM-KNN	0.832	0.931	0.959
SVM	0.821	0.924	0.954
NB	0.766	0.826	0.913

For track regression prediction model, LSTM and KNN algorithm model with a good calibration of the route were used in the experiment to predict the route through AIS data inputted. The experimental statistical results were shown in table8, where the horizontal axis represented various prediction points and the vertical axis represented the prediction accuracy, with the unit being percentage. Observing from the overall trend of figure5, the trend of each curve was gradually increasing, indicating that the accuracy of prediction from T1 to T3 was constantly improving. This was because that the ship track from T1 to T3 was known to be longer and longer, and there were more and more descriptions of ship track. Thus, the prediction was more and more accurate. Looking from the prediction results, the prediction effect of LSTM was the best which was obviously better than that of KNN model. Forecast accuracy in T3 point was over 90%, and prediction effect based on the weighted KNN algorithm was poorer, only when the T3 prediction accuracy was over 80%. The phenomenon occurred mainly because of KNN algorithm only finding the most similar data from historical data rather than LSTM algorithm that could sense the features related to the track of changes.

Table V. The table of KNN and LSTM accuracy comparison

Prediction algorithm	Prediction point		
	T1	T2	T3
KNN-3	0.594	0.721	0.774
KNN-5	0.632	0.764	0.791
LSTM	0.721	0.855	0.901



FigVII. The accuracy comparison of KNN and LSTM

## V. CONCLUSION

In this chapter, a variety of machine learning and deep learning algorithms were introduced and used to predict ship track in a period of time in the future by using AIS data. Through comparing the experimental results of the fourth section, the following conclusions could be the LSTM model on track. Without using traditional machine learning algorithms to forecast in two steps, LSTM not only accelerated the speed of the algorithm model, and avoided the superposition of two step of forecasting error. Therefore, in the term of the implementation steps, the LSTM algorithm on the track of ship AIS data prediction was simple than traditional machine learning algorithms. However, in terms of accuracy of the algorithm, SVM and SVM-KNN are better than NB and LSTM.

When selecting LSTM to identify the risk of the route, it was suggested that 0.85 should be used as the basis to judge whether the route currently selected by the captain was dangerous. Because of the accuracy of prediction, SVM-KNN was selected to identify the danger of the route, this paper suggested using 0.9 as the basis to judge whether the route currently selected by the captain was dangerous.

## REFERENCES

- [1] Ke.Han, Qiang.Xie. Method of filling missing values of ship monitoring data based on improved KNN [J]. Beijing: information and computer (theory edition). 2016(21)
- [2] Qiong.Li, Li.Chen. An improved text classification method for support vector machines [J]. Shaanxi: computer technology and development. 2015(05)
- [3] Xiangyu Ren, Jiansen Zhao. Prediction of ship navigation based on AIS data and LSTM network[J].Shanghai :Journal of Shanghai maritime university.2019(06)
- [4] C.Simon,P.Weber,A.Evsukoff. Bayesian networks inference algorithm to implement Dempster Shafer theory in reliability analysis[J].Reliability Engineering and System Safety,2007(7).
- [5] Liyan Dong, Peng Sui, Peng Sun. A new naive bayesian classification algorithm based on semi-supervised learning [J]. Jilin: journal of jilin university (engineering engineering edition). 2016(03)

**Hong Zhou** Born in Nanjing, Jiangsu Province in June 1974, professor, doctor of engineering, doctoral supervisor, mainly research on Advanced design and manufacturing technology of ship and ocean engineering..

**Qiyi Zhang** Born in Wuxi, Jiangsu Province in March 1994, graduate student, mainly research on Advanced design and manufacturing technology of ship and ocean engineering.