applied sciences

MDPI

Article

Spatio-Temporal Transformer Networks for Inland Ship Trajectory Prediction with Practical Deficient Automatic Identification System Data

Youan Xiao [1], Xin Luo [1], Tengfei Wang [2,3,*] and Zijian Zhang [3]

[1] School of Information Engineering, Wuhan University of Technology, Wuhan 430063, China; youan@whut.edu.cn (Y.X.); luo@whut.edu.cn (X.L.)
[2] State Key Laboratory of Maritime Technology and Safety, Wuhan University of Technology, Wuhan 430063, China
[3] School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China; 333252@whut.edu.cn
[*] Correspondence: wangtengfei@whut.edu.cn

Abstract: Inland waterways, characterized by their complex, narrow paths, see significantly higher traffic volumes compared to maritime routes, increasing the regulatory demands on traffic management. Predictive modeling of ship traffic flows, utilizing real AIS historical data, enhances route and docking planning for ships and port managers. This approach boosts transportation efficiency and safety in inland waterway navigation. Nevertheless, AIS data are flawed, marred by noise, disjointed paths, anomalies, and inconsistent timing between points. This study introduces a data processing technique to refine AIS data, encompassing segmentation, outlier elimination, missing point interpolation, and uniform interval resampling, aiming to enhance trajectory analysis reliability. Utilizing this refined data processing approach on ship trajectory data yields independent, complete motion profiles with uniform timing. Leveraging the Transformer model, denoted TRFM, this research integrates processed AIS data from the Yangtze River to create a predictive dataset, validating the efficacy of our prediction methodology. A comparative analysis with advanced models such as LSTM and its variants demonstrates TRFM's superior accuracy, showcasing lower errors in multiple metrics. TRFM's alignment with actual trajectories underscores its potential for enhancing navigational planning. This validation not only underscores the method's precision in forecasting ship movements but also its utility in risk management and decision-making, contributing significantly to the advancement in maritime traffic safety and efficiency.

Keywords: maritime management; big data; traffic flow prediction; transformer

check for updates

Citation: Xiao, Y.; Luo, X.; Wang, T.; Zhang, Z. Spatio-Temporal Transformer Networks for Inland Ship Trajectory Prediction with Practical Deficient Automatic Identification System Data. *Appl. Sci.* **2024**, *14*, 10494. https://doi.org/10.3390/app142210494

Academic Editor: Koji Murai

Received: 8 March 2024
Revised: 10 June 2024
Accepted: 12 June 2024
Published: 14 November 2024

1. Introduction

Inland waterways are usually more winding and narrower than maritime routes. Due to the limitations of the inland geographical environment, ships navigating in inland waterways require higher technical standards and risk management capabilities, making it more challenging than maritime navigation. Moreover, inland ships have limited navigation routes, and inland waterways often connect important cities and ports, carrying a large volume of cargo transport tasks. This leads to higher traffic density, more ships, and more frequent transportation activities in inland shipping. Faced with the busyness and complexity of inland navigation, traffic management departments need to bear greater regulatory pressure and responsibility to ensure the safe passage of ships and prevent accidents. Ship traffic flow prediction can analyze the navigation status of ships in future periods. Based on this analysis, ship and port managers can better plan the navigation routes of ships, select the optimal routes to reduce travel time and fuel consumption, and improve the efficiency of ship transportation. Additionally, it can help managers predict port congestion and the availability of docking areas, thus allowing for more rational arrangements of ship docking time and location. This reduces waiting times and alleviates congestion, improving port throughput efficiency. Ship traffic flow prediction can also identify potential collision risks, traffic congestion areas, and unsafe waterways in advance, helping ship and port managers take appropriate measures to avoid accidents and enhance the safety of ship transportation. The AIS system is a technology widely used for automatic ship identification and tracking. Through AIS, real-time data such as the location, speed, and course of ships can be collected and recorded. By statistically analyzing and data mining AIS historical data, the characteristics and patterns of ship traffic flow in inland waterways can be extracted, which can be used to establish models for predicting inland ship traffic flow. Furthermore, analyzing and modeling ship traffic flow using AIS historical data can continuously optimize prediction algorithms and models. With the collection of more real-time data, the prediction models can be updated and improved continuously, enhancing the accuracy and precision of predictions.

With the development of the maritime economy, intelligent ships have become one of the key research directions in the field of maritime transportation. Intelligent ships can enhance the safety, environmental friendliness, and cost-effectiveness of maritime vessels, while also providing support for activities such as maritime rescue, waterway monitoring, and maritime security [1]. The real-time perception, prediction, and analysis of information related to ship positions, statuses, and behaviors are crucial foundations for ensuring the safety of intelligent ship navigation. AIS data (Automatic Identification System data) serve as a vital source of information in this context. It consists of messages automatically sent and received by vessels through the Automatic Identification System, encompassing static data, dynamic data, and navigation-related data. AIS data can reveal essential details about a vessel's identity, location, speed, direction, destination, and more, playing a critical role in the monitoring and management of maritime traffic [1]. Therefore, by employing mathematical models and algorithms, it is possible to predict the future positions of vessels within a certain time frame based on historical and real-time AIS data. This prediction can offer valuable references and support for intelligent navigation, collision avoidance decisions, energy efficiency optimization, safety warnings, and other related applications [2].

In response to the challenges in AIS data processing and the limitations of current ship trajectory prediction models, this paper makes the following three main contributions:

Addressing the real-time requirements for trajectory prediction with AIS data, we propose an AIS trajectory data preprocessing method. Our method includes trajectory segmentation, removal of anomalous/redundant points, patching missing points, and resampling at equal time intervals. It is capable of handling large volumes of AIS data, generating equidistant and complete ship trajectories, meeting the input requirements of trajectory prediction models.

To overcome the shortcomings of recurrent neural networks in ship trajectory prediction, we introduce a ship trajectory prediction method based on the TRFM (Transformer) model. Leveraging the multi-head attention mechanism of the TRFM model, it adapts to focus on features throughout the sequence and assigns higher weights to crucial trajectory points. By extracting global temporal feature information from the trajectory sequence based on attention weights, it addresses the issue of traditional recurrent neural networks primarily attending to the tail-end features of the sequence and overlooking global sequence characteristics. Additionally, the TRFM model performs parallel computation when processing sequence features, enhancing the efficiency of trajectory prediction.

To ensure consistent trajectory prediction performance across different latitude and longitude regions and enhance model generalization, we propose a trajectory prediction data structure where the input and output sequences are based on the relative values of latitude and longitude. Differential processing is applied to adjacent trajectory points' latitude and longitude, utilizing the heading angle, velocity, relative longitude, and relative latitude as input sequence information, and relative longitude and relative latitude as output sequences. The final latitude and longitude positions of the trajectory sequence are recorded for regressing the absolute latitude and longitude values. This approach improves model generalization.

The remaining sections of this paper are organized as follows: Section 2 focuses on the detailed aspects of AIS data processing and the design of the TRFM model structure. Section 3 presents experimental comparative analyses. Section 4 provides the conclusion and Section 5 presents the limitations and future work.

2. Literature Review

Due to issues related to receiving equipment, transmitting devices, and network transmission, raw AIS data often contain some level of noise, necessitating preprocessing for effective utilization. Numerous efforts have been made by researchers in the field of AIS data processing. Xue proposed a trajectory similarity measurement and clustering method to divide scenes into semantic regions, aiming to mitigate the negative effects caused by AIS [3]. Qiao conducted research on the automated processing of AIS data and developed an automated processing platform [4]. This platform achieved functionality for updating AIS data, vessel static information, route information, and voyage information. Guo proposed an anomaly detection method based on AIS trajectories. They integrated the kinematic information of vessels in AIS data and reduced trajectory anomalies through data preprocessing, motion estimation, and error clustering [5]. Zhao introduced a preprocessing model consisting of data cleaning, trajectory extraction, and trajectory compression stages. This method effectively removes excess noise and reduces the size of trajectory data [6]. Liu proposed preprocessing methods such as navigation data extraction, abnormal data handling, and missing data interpolation to address issues such as AIS data loss, inaccuracy, and incomplete preservation of dynamic navigation features [7]. Bakht presented a two-step AIS processing method involving trajectory interpolation and the application of trajectory point detection methods to interpolated and observed AIS messages [8]. This approach enhances trajectory resolution effectively. Guo proposed an unsupervised knowledge mining framework for generating maritime traffic networks [9]. This framework integrates AIS data preprocessing algorithms, enabling the generation of high-quality, spatio-temporally continuous maritime trajectory data from low-quality inputs.

While the aforementioned methods demonstrate good performance in processing individual AIS data values for single vessels, they are time-consuming and lack real-time capabilities when handling a large volume of ship data simultaneously. Moreover, they often do not address the issue of uneven time intervals between trajectory points, making them less suitable for the input requirements of trajectory prediction models.

Traditional trajectory prediction models can be primarily categorized into two classes: statistical learning methods and machine learning methods. Deep learning, as a branch of machine learning, has gained prominence in recent years, and deep learning-based prediction methods have become the mainstream in trajectory prediction research. Wang proposed a deep learning prediction model based on the Bi-GRU network, which outperformed LSTM and standard GRU network prediction models in terms of lower error and higher accuracy [10]. Liu proposed an optimized Attention–LSTM neural network for dynamic navigation prediction, and validated the accuracy and robustness of the model through a simulation analysis [7]. The results demonstrate that this method achieves the high-precision prediction of vessel longitude, latitude, heading, and speed. Capobianco extended the deep learning framework for trajectory prediction tasks by exploring how recurrent encoder–decoder neural networks can not only predict but also generate corresponding prediction uncertainty for both aleatoric and epistemic uncertainties using a Bayesian model [11]. Zhang introduced a novel trajectory prediction model called PESO, comprising parallel encoders, ship-oriented decoders, and Semantic Location Vectors (SLVs) [12]. Parallel encoders aim to capture more information in feature representations, while ship-oriented decoders use SLV to guide predictions and better represent the spatial correlations of historical trajectory points. Gao presented a high-precision multi-step prediction method called MP-LSTM, which combines current trajectory data and historical trajectory data [13]. It uses LSTM neural network models to predict trajectory support points and then performs multi-step predictions using cubic spline interpolation for start, support, and endpoint prediction. This approach achieves high-precision predictions for both short and long distances while accurately predicting positions at each time step. Liu proposed an improved Attention–LSTM model for dynamic navigation prediction, incorporating an Attention module for differential weighting and feature extraction from navigation sequences [7]. The LSTM module integrates information for final dynamic predictions. Feng introduced the IS-STGCNN, a spatio-temporal graph convolutional model for trajectory prediction [14]. This model accounts for vessel interactions' impact on trajectory prediction, uses social sampling to update node vector representations, and employs MPC for prediction result correction. Liu introduced a Hybrid Driven Framework, which fuses data-driven predictors with vessel motion constraints using a linear Kalman filter [15]. It uses a learning-based LSTM network as a data-driven predictor and employs the Kalman filter's physical-driven module to estimate future vessel trajectories using the data-driven predictor's sequential output as continuous measurements. Liu proposed an LSTM-based interactive ship trajectory prediction framework known as QSD-LSTM, embedding Quaternion Ship Domain (QSD) [16]. QSD aids in avoiding unnecessary collisions between adjacent vessels during the trajectory prediction process. Mehri proposed a Context-aware Long Short-Term Memory (CLSTM) network [17]. Compared to the Long Short-Term Memory (LSTM) network, this framework improves prediction accuracy by 15.31%. Han proposed a deep generative model based on the conditional variational autoencoder framework to learn ship motion and predict future trajectories [18]. Experimental results indicate that this model outperforms baseline methods, including both kinematics-based and data-driven approaches. Chen et al. [19] presented a trajectory prediction method based on Bi-GRU and trajectory direction vectors (TDVs) with an attention mechanism. They constructed a TDV, which correlates latitude and longitude with heading and speed, and used an adaptive attention mechanism to eliminate the influence of unreasonable predicted trajectory points in the trajectory. Wang proposed a Deep Attention-Aware Spatio-Temporal Graph Convolutional Network (DAA-SGCN) based on AIS data to predict future vessel trajectories [20]. It mainly consists of three modules: the vessel trajectory motion information encoding, spatio-temporal feature extraction module, and trajectory prediction module. Based on extensive experiments, compared to the optimal baseline model, the predictive performance on *ADE* and *FDE* metrics improved by 74% and 69%, respectively. Slaughter proposed a simple fusion-based RNN method for predicting vessel trajectories [21]. This approach can easily integrate other temporal features. Experimental results demonstrate that this method achieves state-of-the-art performance in three major coastal regions of the United States. When predicting the next three hours, it outperforms competing methods by 0.88 km.

Li et al. [22] extracted five machine learning methods and seven deep learning methods from state-of-the-art literature reviews. These methods were employed to perform AIS data-based trajectory predictions in three representative busy coastal waters and compare their prediction performance in real-world scenarios. The performance of all twelve methods was evaluated and analyzed, providing a detailed exploration of the characteristics and effectiveness of these trajectory prediction methods. However, the paper did not conduct an in-depth optimization of the twelve aforementioned models. This is uncommon in practical applications. Jiang et al. [23] integrated the LSTM structure into the deep learning Transformer algorithm framework to address the limitations of LSTM in capturing long-distance sequence information, thereby achieving the complementary advantages of long-range dependencies in temporal and spatial features.

In ship trajectory prediction tasks, most researchers have employed traditional recurrent neural networks as the model foundation. Although recurrent neural networks are proficient at learning the temporal characteristics of ship trajectory sequences, their model inference processes are typically sequential [24]. This means that calculating the feature points for the next sequence requires waiting for the completion of the computation of the previous feature point. Additionally, recurrent neural networks are prone to losing information from the beginning of long trajectory sequences. Therefore, in the proposed method of this paper, we have also targeted to add a multi-head attention mechanism. Compared to traditional recurrent neural networks (RNNs), where features are passed sequentially at each time step, the multi-head attention mechanism allows the features at each time step to perceive the global temporal features and their relationship with the current node data. The advantage of multi-head attention is the ability to consider the effect of global temporal features on the current node features from different dimensions.

3. Methodology

The main framework for ship trajectory prediction based on the TRFM model, as proposed in this paper, is illustrated in Figure 1. The data preprocessing section involves the extraction of ship trajectories, removal of anomalous and redundant points, imputation of missing points, equidistant time interval resampling, and dataset construction, all performed on the raw collected ship AIS data. Through this preprocessing procedure, the original AIS data become more robust, effectively addressing issues such as data anomalies, missing data, and discontinuous time intervals caused by network transmission, reception, and transmission. The inference section utilizes the TRFM model to predict ship trajectories. The TRFM model adopts an encoder–decoder structure, with its core component being the multi-head attention mechanism. It is able to effectively extract global spatio-temporal features of trajectories, thereby enhancing the accuracy of trajectory prediction. This mechanism effectively enhances the feature correlations between sequence points and enables parallel processing when handling temporal tasks, greatly improving the model's computational speed [25]. The model prediction uses the differential relative values of latitude and longitude as inputs, and the predicted results are obtained by regressing the absolute values of latitude and longitude to derive the final predicted trajectory. This can make the model's trajectory prediction more generalizable, rather than limited to trajectory predictions within the region where the training data are located. By constructing a trajectory prediction dataset and training and predicting the model on this dataset, the performance of the model in trajectory prediction tasks is compared with other state-of-the-art trajectory prediction models. Comprehensive evaluations of the model's prediction performance are conducted using metrics such as *MAE*, *RMSE*, *MSE*, *FDE*, and *ADE*.

3.1. Definitions and Problems

3.1.1. Definitions

Definition 1. *Ship Trajectory. A ship trajectory consists of a series of timestamped trajectory points, denoted as* $Traj = \{Po_1, \cdots, Po_i, \cdots, Po_L\}$, $Traj \in \mathbb{R}^{N_L \times N_D}$, *where* N_L *represents the number of trajectories, and* N_D *represents the feature dimensions. Each trajectory point* $Po_i = \{cog_i, sog_i, lon_i, lat_i, ns_i, t_i\}$, $i = 1, \cdots, L$, $Po_i \in \mathbb{R}^{N_D}$ *represents the ship's attributes at each timestamp. Here,* cog_i, sog_i, lon_i, lat_i, ns_i, *and* t_i, *respectively, represent the course over ground, speed over ground, longitude, latitude, navigation states, and timestamp.*

Definition 2. *Ship Trajectory Dataset. Let* $TD = \{D^1, D^2, D^3, \cdots, D^K\}$, $TD \in \mathbb{R}^{N_K \times N_C \times N_D}$, *where* N_K *represents the number of ship trajectory prediction combinations. Each trajectory prediction combination is represented as* $D^j = \{X^j, Y^j, Z^j\}$, $X^j \in \mathbb{R}^{N_H \times N_{D4}}$, $Y^j \in \mathbb{R}^{N_P \times N_{D2}}$, $Z^j \in \mathbb{R}^{N_{D2}}$. *Here,* X^j *represents the input trajectory sequence, where* N_H *represents the input sequence length and* N_{D4} *represents the feature dimensions (4 in this case).* Y^j *represents the output trajectory sequence, where* N_P *represents the output sequence length and* N_{D2} *represents the feature dimensions (2 in this case).* Z^j *represents the coordinates of the last trajectory point of the input trajectory sequence.*

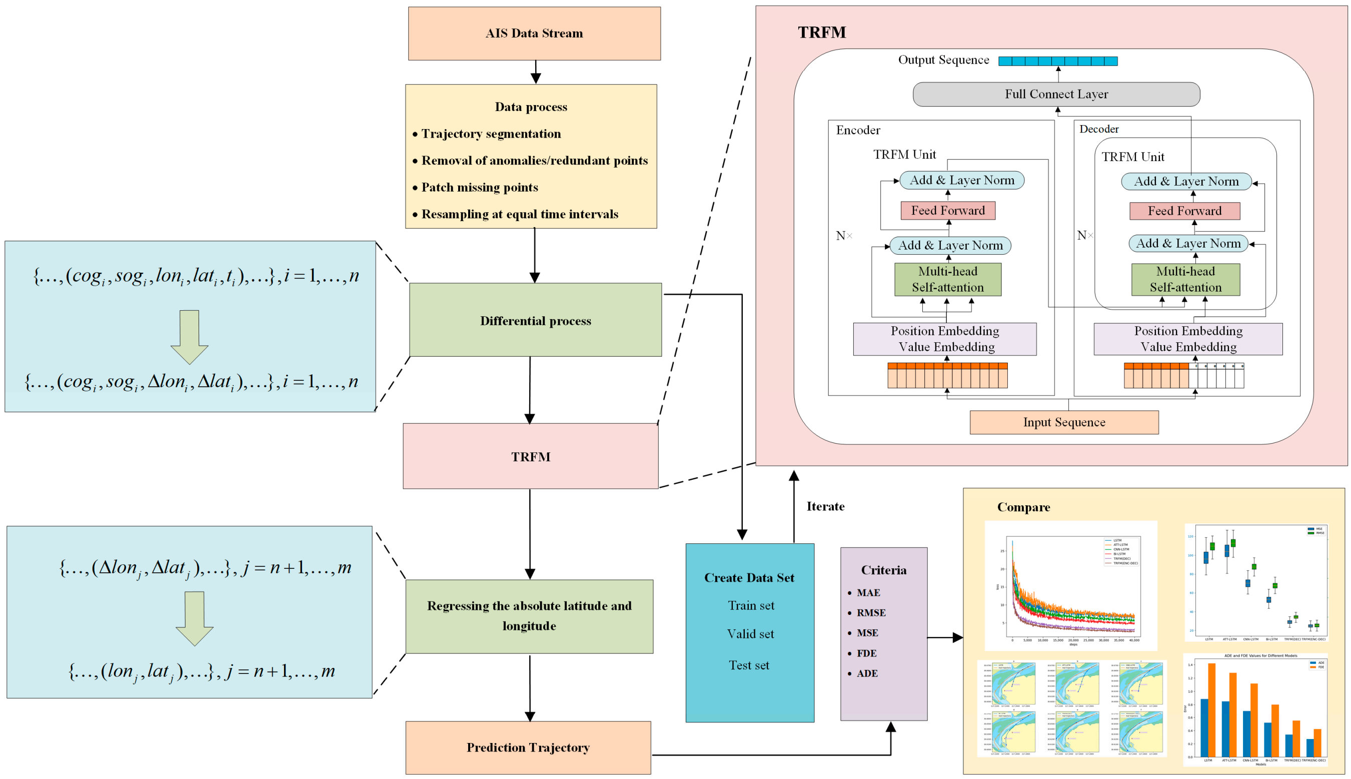
AIS Data Stream
Data process
• Trajectory segmentation
• Removal of anomalies/redundant points
• Patch missing points
• Resampling at equal time intervals
$\{\ldots,(cog_i,sog_i,lon_i,lat_i,t_i),\ldots\},i=1,\ldots,n$
$\{\ldots,(cog_i,sog_i,\Delta lon_i,\Delta lat_i),\ldots\},i=1,\ldots,n$
Differential process
TRFM
$\{\ldots,(\Delta lon_j,\Delta lat_j),\ldots\},j=n+1,\ldots,m$
$\{\ldots,(lon_j,lat_j),\ldots\},j=n+1,\ldots,m$
Regressing the absolute latitude and longitude
Prediction Trajectory
TRFM
Output Sequence
Full Connect Layer
Encoder
TRFM Unit
Add & Layer Norm
Feed Forward
Add & Layer Norm
Multi-head Self-attention
N×
Position Embedding Value Embedding
Decoder
TRFM Unit
Add & Layer Norm
Feed Forward
Add & Layer Norm
Multi-head Self-attention
N×
Position Embedding Value Embedding
Input Sequence
Iterate
Create Data Set
Train set
Valid set
Test set
Criteria
• MAE
• RMSE
• MSE
• FDE
• ADE
Compare

Figure 1. Architecture of ship trajectory prediction method based on TRFM model.

3.1.2. Problems

Origin of the Problems: In the evaluation of ship navigation safety, it is crucial not only to obtain accurate ship trajectories but also to predict the future trajectory direction of ships. This is of significant importance for assessing future ship risks and making proactive decisions. The main problems studied and discussed in this paper are as follows:

Problem 1. *How to accurately predict the ship's trajectory for a certain period in the future?*

To accurately predict the ship's trajectory for a certain period in the future, the ship trajectory data are first processed (φ_d) to obtain trajectory data suitable for the input structure of the model. Then, this processed ship trajectory data are fed into the trajectory prediction model (f_m) to obtain the ship's predicted trajectory for the future. The formula is shown in (1). The trajectory prediction is expressed as shown in Figure 2.

$$\{Po^j_{n+1}, Po^j_{n+2}, \cdots, Po^j_m\} = f_m(\varphi_d(\{Po^j_1, Po^j_2, \cdots, Po^j_n\}, \pi), \theta) \tag{1}$$

Problem 2. *How to measure the effectiveness of ship trajectory prediction models?*

To evaluate the ship trajectory prediction model (f_m), Mean Absolute Error (MAE) is utilized for optimization, shown in Formula (2). Mean Squared Error (MSE), Root Mean Square Error ($RMSE$), Average Displacement Error (ADE), and Final Displacement Error (FDE) are used to assess the accuracy of the trajectory prediction model.

$$\text{Optimal } f_m \leftarrow \min(MAE) \tag{2}$$

$$\text{Eualuate } f_m \leftarrow \text{mean}\{MSE, RMSE, ADE, FDE\} \tag{3}$$

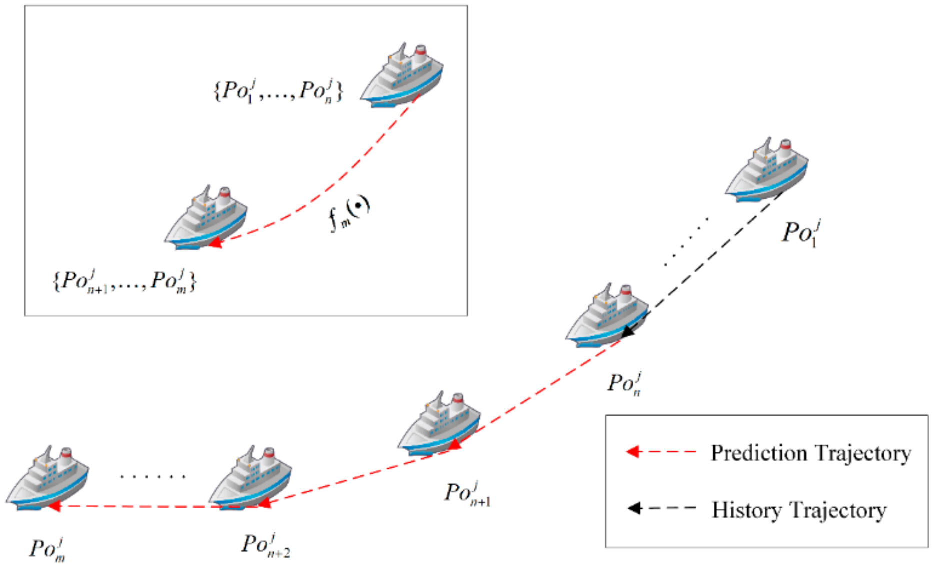

Figure 2. A representation of the trajectory prediction problem.

3.2. Data Processing

The data packets transmitted and received for ship AIS data contain information about the ship's identity, position, speed, heading, and other details. AIS data can be categorized into two main types: dynamic information and static information. Dynamic information includes the ship position, course over ground, speed over ground, navigation status, timestamp, and more. It is automatically updated by the AIS transponder's position sensor, with the transmission interval adjusting automatically based on changes in ship speed and heading. Typically, dynamic information is transmitted every 3 to 5 s. Static information includes the IMO number, call sign, MMSI number, vessel name, ship type, dimensions, and other details. This information is manually input by users and is transmitted at a longer interval, usually every 6 min.

For this research, we have chosen AIS data from the Yangtze River Basin on 17 March 2023 as our study object. This region is one of China's most important inland waterway transportation areas, featuring an extensive network of waterways and a high volume of vessel traffic. Moreover, the complex river channels, hydrology, and navigation challenges in this area make it an ideal case for evaluating ship trajectory prediction performance. Specific AIS data fields are shown in Table 1, listing only a portion of the data fields for reference.

Table 1. Structure of Raw AIS Data.

MSGTIME	MMSI	LON	LAT	SOG	COG	NS
17 March 2023 17:51:10	413xxx567	114.2015	30.44428	0	354	1
17 March 2023 17:51:10	413xxx965	114.1501	30.59679	0	331	1
17 March 2023 17:51:10	413xxx381	114.0261	30.22972	5.2	311.9	0
17 March 2023 17:51:10	413xxx688	114.4845	30.67789	0	342	1
17 March 2023 17:51:10	413xxx182	114.4793	30.67895	2	251.3	0
17 March 2023 17:51:10	413xxx371	114.0638	30.09095	3.9	242.6	0

The following will provide a detailed explanation of each part of the AIS trajectory data processing method proposed in this paper.

3.2.1. Trajectory Segmentation

Based on the AIS data structure, AIS data are divided into several trajectory sets according to the unique ship identification number, MMSI, represented as $Traj^{mmsi}$. Since AIS data for each ship consist of multiple segments of voyage data, it is necessary to segment different voyages for the same ship. This paper performs trajectory segmentation at two different levels of granularity based on ship status and ship speed from the AIS data. Trajectory segmentation is expressed as shown in Figure 3.

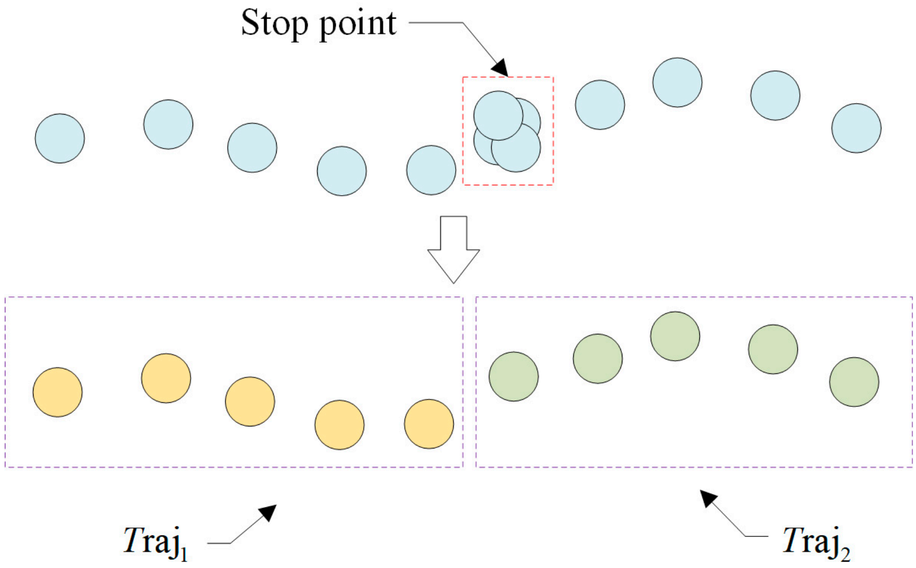

Figure 3. Trajectory segmentation.

Firstly, the trajectories in the AIS data are divided into navigation states (NSs). In the method proposed in this paper, NSs mainly include navigation and anchoring, represented by 0 and 1, respectively. Based on these two states, the original set of navigation trajectories can be divided into multiple navigation trajectory segments. Then, further divide the trajectories based on the ship's speed. When trajectory points with speed $sog_i = 0$ and $ns_i = 1$ appear, mark them as stopping points (Po_{stop}) for new sailing segments. Then, cut the trajectory based on Po_{stop}, and the cut trajectories $\{Traj_1, \cdots, Traj_n\}$ are added to the navigation trajectory set, represented as $Traj^{nav}$. This results in finer-grained trajectories.

$$Po_{stop} = \{Po_i | sog_i = 0, ns_i = 1\} \tag{4}$$

Since trajectories with a smaller number of points have less significance, trajectories in the new trajectory set with point counts less than the specified threshold length l_{min} are removed; we obtain $Traj^{nav}$ as Formula (5). The final set of ship trajectories serves as the foundation for the subsequent processing, analysis, and dataset construction.

$$\{Traj_j | len(Traj_j) \geq l_{min}\} \not\subset Traj^{nav}, j = 1, \cdots, n \tag{5}$$

3.2.2. Removal of Anomalies/Redundant Points

Biases, errors, and duplicate data may occur during the collection, transmission, and network latency of AIS data, and there is the fact that adjacent data points in AIS data do not have equal time intervals, which is not conducive to the prediction and analysis of subsequent models. Therefore, in addition to removing abnormal data, this paper also removes some redundant points with relatively small changes in navigation states and points with significant changes in navigation states, collectively referred to as removal points (Po_{drop}).

This paper calculates the change rate of course over ground (CRC) and change rate of speed over ground (CRS) between adjacent points. Minimum and maximum thresholds for CRC and CRS are set. Points with values less than the minimum threshold indicate insignificant changes in navigation states and are removed. Points with values greater than the maximum threshold indicate abnormal trajectory points and are also removed. The calculation of crc_i and crs_i for each trajectory point is shown in Formula (6).

$$\begin{bmatrix} crc_i \\ crs_i \end{bmatrix} = \begin{bmatrix} cog_{i+1} - cog_i \\ sog_{i+1} - sog_i \end{bmatrix} [t_{i+1} - t_i]^{-1} \tag{6}$$

We calculate the values of crc_i and crs_i for each trajectory point and perform mathematical statistics on them. We use a Gaussian distribution denoted as 'a' to fit the distribution of global CRC and CRS. A simplified representation is as follows:

$$\overline{x}^a = \frac{1}{n}\sum_{i=1}^{n} x_i^a \tag{7}$$

$$(\sigma^a)^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i^a - \overline{x}^a)^2 \tag{8}$$

where $\overline{x}^a \in R^2$ represents the mean of the samples, $n \in R$ is the total sample count, $a = \{CRC, CRS\}$, and $(\sigma^a)^2 \in R^2$ represents the variance of the samples. Based on the distribution statistics, we use the value of 3σ as the threshold boundary for CRC and CRS, and their respective mathematical expressions are given by (9). b_{min}^{crc} and b_{max}^{crc} represent the minimum and maximum values of the CRC threshold boundary, while b_{min}^{crs} and b_{max}^{crs} represent the minimum and maximum values of the CRS threshold boundary.

$$\begin{bmatrix} b_{min}^{crc} & b_{max}^{crc} \\ b_{min}^{crs} & b_{max}^{crs} \end{bmatrix} = \begin{bmatrix} \overline{x}_1^a \\ \overline{x}_2^a \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} + 3 \begin{bmatrix} (\sigma_1^a)^2 \\ (\sigma_2^a)^2 \end{bmatrix} \begin{bmatrix} -1 & 1 \end{bmatrix} \tag{9}$$

For each trajectory in the navigation trajectory set ($Traj^{nav}$), we traverse and remove discovered anomaly points and redundant points. For each trajectory point in the set, the points to be removed are denoted as $Po_{drop} = \{Po_{er}, Po_{co}, Po_{de}, Po_{re}\}$.

- Error Points (Po_{er}): Trajectory points with cog_i, sog_i, lon_i, and lat_i values outside reasonable ranges, mathematically defined in Formula (10).
- Coincident Points (Po_{co}): Points in the trajectory where sog_i is greater than 0, but the coordinates (lon_i, lat_i) are the same. Mathematically defined in Formula (11).
- Deviation Points (Po_{de}): Current trajectory points that have a significant offset from adjacent trajectory points, mathematically defined in Formula (12).
- Redundant Points (Po_{re}): Current trajectory points where crc_i and crs_i with the previous trajectory point are less than the minimum threshold or greater than the maximum threshold, mathematically defined in Formula (13).

$$Po_{er} \neq \{Po_i | 0 \leq cog_i \leq 360, 0 \leq sog_i \leq 25, 0 \leq lon_i \leq 180, 0 \leq lat_i \leq 90\} \tag{10}$$

$$Po_{co} = \{Po_i | sog_i > 0, lon_i = lon_j, lat_i = lat_j, i \neq j\} \tag{11}$$

$$Po_{de} = \{Po_i | \frac{dist(Po_{i+1}, Po_i)}{t_{i+1} - t_i} \geq V_{\max}\} \tag{12}$$

$$Po_{re} \neq \{Po_i | b_{min}^{crc} \leq crc_i \leq b_{max}^{crc}, b_{min}^{crs} \leq crs_i \leq b_{max}^{crs}\} \tag{13}$$

After the removal of anomalous and redundant values, the trajectory data in the collection have been significantly compressed, while ensuring that the original trajectory remains undistorted, as many unnecessary trajectory points as possible have been removed. This greatly preserves valid trajectory points that represent the vessel's motion state.

3.2.3. Patching Missing Points

After the removal of anomalies/redundant points from the trajectory collection, there will be a significant number of missing trajectory points. In order to make the trajectory complete, we need to patch in these missing trajectory points. Since the previous processing steps retained enough trajectory points representing the vessel's motion state, the missing trajectory points that need to be inserted are mostly associated with relatively stable uniform motion states. Therefore, a simple linear interpolation method is used to fill in the missing trajectory points with evenly spaced intervals in time. When the time interval between two adjacent trajectory points, Po_i and Po_j, is greater than the specified time interval threshold $T_{interp} \in \mathbb{R}$, the linear interpolation method is used to interpolate five state variables of the missing trajectory points between Po_i and Po_j. These variables include the heading angle (cog_i), speed over ground (sog_i), longitude (lon_i), latitude (lat_i), and timestamp (t_i). The number of inserted points is denoted as $n \in \mathbb{N}$.

$$n = \left\lfloor \frac{t_j - t_i}{T_{interp}} \right\rfloor \tag{14}$$

We use linear interpolation to patch in the missing points for the five state variables of the missing trajectory points. The insertion formula is as follows:

$$\begin{bmatrix} cog_{i+k} \\ sog_{i+k} \\ lon_{i+k} \\ lat_{i+k} \\ t_{i+k} \end{bmatrix} = \begin{bmatrix} cog_i \\ sog_i \\ lon_i \\ lat_i \\ t_i \end{bmatrix} + kT_{interp}[t_{i+k} - t_i]^{-1} \begin{bmatrix} cog_j - cog_i \\ sog_j - sog_i \\ lon_j - lon_i \\ lat_j - lat_i \\ t_j - t_i \end{bmatrix}, k = 1, 2, 3 \ldots N \tag{15}$$

After the imputation of missing points, the time interval between trajectory points with minimal changes in navigation state in the trajectory collection is T_{interp}. Only a small portion of segments with drastic changes in vessel state still maintain non-uniform time intervals.

3.2.4. Resampling at Equal Time Intervals

After the missing point imputation, the obtained trajectory data are relatively complete, and the time intervals of inserted points are all equal. However, trajectory points with frequent state changes still do not have equidistant sampling intervals. These data are not conducive for subsequent model training and prediction. Therefore, it is necessary to perform equidistant resampling on the trajectory data, setting the sampling interval as T_{samp}. In order to prevent excessive loss of original information in the equidistant trajectory data, this paper uses both small time interval T_s and linear interpolation, as well as a large time interval T_l and method to obtain equidistant trajectory data. The definitions of small and large time intervals are as follows:

$$\begin{bmatrix} T_s \\ T_l \end{bmatrix} = \begin{bmatrix} \frac{1}{p} \\ 1 \end{bmatrix} T_{samp}, p \in \mathbb{N} \tag{16}$$

where 'p' represents the scaling factor for large and small time intervals. First, the trajectories in the trajectory collection are traversed. When a trajectory point with a time interval less than T_s compared to the previous trajectory point is encountered, no processing is performed, and the traversal continues. If a point with a time interval greater than or equal to T_l is encountered, linear interpolation is used to interpolate cog_i, sog_i, lon_i, and lat_i onto equidistant trajectory points. Finally, a large time interval resampling is performed on the trajectory collection. As a result, the time interval between each trajectory point in the trajectory collection is T_{samp}, ensuring uniformity and stability of the trajectory points, ensuring consistency between the data, and making a trajectory prediction analysis easier.

3.2.5. Dataset Construction

After the above data processing, the time intervals between each trajectory point are equal. When constructing the dataset, it is necessary to compute the relative values $\{lon_i, lat_i\}$ of the longitude and latitude of each trajectory point with respect to the previous trajectory point $\{\Delta lon_i, \Delta lat_i\}$. The differential values of longitude and latitude, along with the current true heading and ground speed, constitute the features of the current trajectory point. In the trajectory prediction model of this paper, since the time intervals between the points in the trajectory sequence are equal, the input sequence omits the time dimension. The input data have 4 dimensions, namely the true heading angle, ground speed, relative longitude, and relative latitude. The output sequence data have 2 dimensions, namely relative longitude and relative latitude. There is the input sequence point $x_i^j \in \mathbb{R}^{N_{D4}}$, as shown in mathematical expression (17).

$$x_i^j = \left\{cog_i^j, sog_i^j, \Delta lon_i^j, \Delta lat_i^j\right\} \tag{17}$$

The predicted trajectory sequence consists of differential values of longitude and latitude, with the output sequence point $y_i^j \in \mathbb{R}^{N_{D2}}$, as shown in mathematical expression (18).

$$y_i^j = \left\{\Delta lon_i^j, \Delta lat_i^j\right\} \tag{18}$$

At the same time, it is necessary to record the reference point used for absolute value calculation, which is the longitude and latitude coordinates of the last point in the input sequence, as shown in Formula (19), and n presents the final position of history trajectory.

$$Z^j = \left\{lon_n^j, lat_n^j\right\} \tag{19}$$

Combining X^j, Y^j, and Z^j forms one input–output sequence pair D^j, as shown in Formulas (20)–(22).

$$X^j = \left\{x_1^j, x_2^j, x_3^j \ldots x_n^j\right\} \tag{20}$$

$$Y^j = \left\{y_1^j, y_2^j, y_3^j \ldots y_m^j\right\} \tag{21}$$

$$D^j = \left\{X^j, Y^j, Z^j\right\} \tag{22}$$

The above is the specific method for processing ship trajectories, which includes the data processing flow and algorithm pseudo-code as shown in Algorithm 1. The purpose of this trajectory prediction data processing algorithm is to preprocess the original AIS ship data and construct a dataset, as well as to form an input structure suitable for ship trajectory prediction models.

Algorithm 1: Data processing algorithms

Input: *init_ais*: Represents the original input AIS data. l_{min}: Represents the min length of trajectory. T_{interp}: Represents the interpolation time interval. T_{samp}: Represents the resample time interval. H: Represents the input sequence length. P: Represents the output sequence length.

Output: Trajectory prediction dataset TD.

Initialization: *MMSI_Traj*: Used to store the trajectory of each unique MMSI. *All_Traj*: Used to store ship trajectories during the processing.

Function: $Repair(\cdot)$: Used to store ship trajectories during the processing. $Resample(\cdot)$: Used to store ship trajectories during the processing.

Process:

```
//Step 1 Trajectory segmentation, removal of outliers and redundant points, patching of missing points, and equidistant time interval resampling.
For $ais_i$ in $init\_ais$ do
    Find the MMSI trajectory that $ais_i$ belongs to in $MMSI_{Traj}$
    $Po_m \leftarrow \{cog_i, sog_i, lon_i, lat_i, ns_i, t_i\}$ from $ais_i$
    Add $Po_m$ to $Traj_j^{mmsi}$
End For
For $Traj_j$ in $MMSI\_Traj$ do
    $Tmp\_Traj \leftarrow$ Create an empty set
    For $Po_i$ in $Traj_j$ do
        Add $Po_i$ to $Tmp\_Traj$
        If current $Po_i$ is stop point do
            If the length of $Tmp\_Traj \geq l_{min}$ do
                Add $Tmp\_Traj$ to $ALL\_Traj$
            Clear the set $Tmp\_Traj$
    End For
End For
For $Traj_j$ in $All\_Traj$ do
    For $Po_i$ in $Traj_j$ do
        If $Po_i$ is removal points do
            remove $Po_i$ from $Traj_j$
    End For
    $Traj_j \leftarrow Repair\left(Traj_j,\ T_{interp}\right)$
    $Traj_j \leftarrow Resample\left(Traj_j,\ T_{sample}\right)$
End For
// Step 2 Differential processing and create dataset.
For $Traj_j$ in $ALL\_Traj$ do
    $Tmp\_Diff \leftarrow$ Create an empty set
    For $Po_i$ in $Traj_j$ do
        $\Delta lon_i,\ \Delta lat_i \leftarrow$ Calculate the latitude and longitude differences between $Po_i$ and $Po_{i-1}$
        $cog_i,\ sog_i \leftarrow$ the heading angle and speed in $Po_i$
        Add $[cog_i, sog_i, \Delta lon_i, \Delta lat_i]$ to $Tmp\_Diff$
    End For
    For m = 1 to M do
        $X \leftarrow Tmp\_Diff$[m:m + H]
        $Y \leftarrow Tmp\_Diff$[m + H:m + H + P]
        $Z \leftarrow Traj_j$[m]
        Add $[X, Y, Z]$ to TD
    End For
End For
```

3.3. TRFM Trajectory Prediction Model

The TRFM trajectory prediction model consists of two parts: the encoding layer and the decoding layer. The structure of the model is shown in Figure 1, where both the encoder and decoder cores are composed of TRFM units. Each TRFM unit comprises two sub-layers: Self-Attention and Feed Forward. Self-Attention calculates the dependency between each position in the sequence and all other positions, while Feed Forward further enhances feature representation while maintaining consistent output dimensions with the input. In this paper, we use differential values for longitude and latitude instead of directly using them as the model's input and output. The output's differential values are calculated through the absolute regression of latitude and longitude, yielding the true predicted trajectory coordinates.

3.3.1. TRFM Unit

The core of the Self-Attention mechanism involves calculating the weighted relationships between the current sequence point and all other sequence points using three vectors: Q (query), K (key), and V (value). The Self-Attention calculation structure is shown in Figure 4. The dot product of Q and K vectors represents the degree of correlation between features of two temporal points. The resulting attention weights, obtained through a $softmax$ layer, are used to weight and sum the features of the current temporal point with those of other temporal points. In each TRFM unit, the Q, K, and V vectors of each temporal feature point can be combined into their respective Q, K, and V matrices. This allows for parallel computation when calculating attention weights, thereby improving model efficiency. The attention calculation formula is shown in Equation (23).

$$Self - Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{23}$$

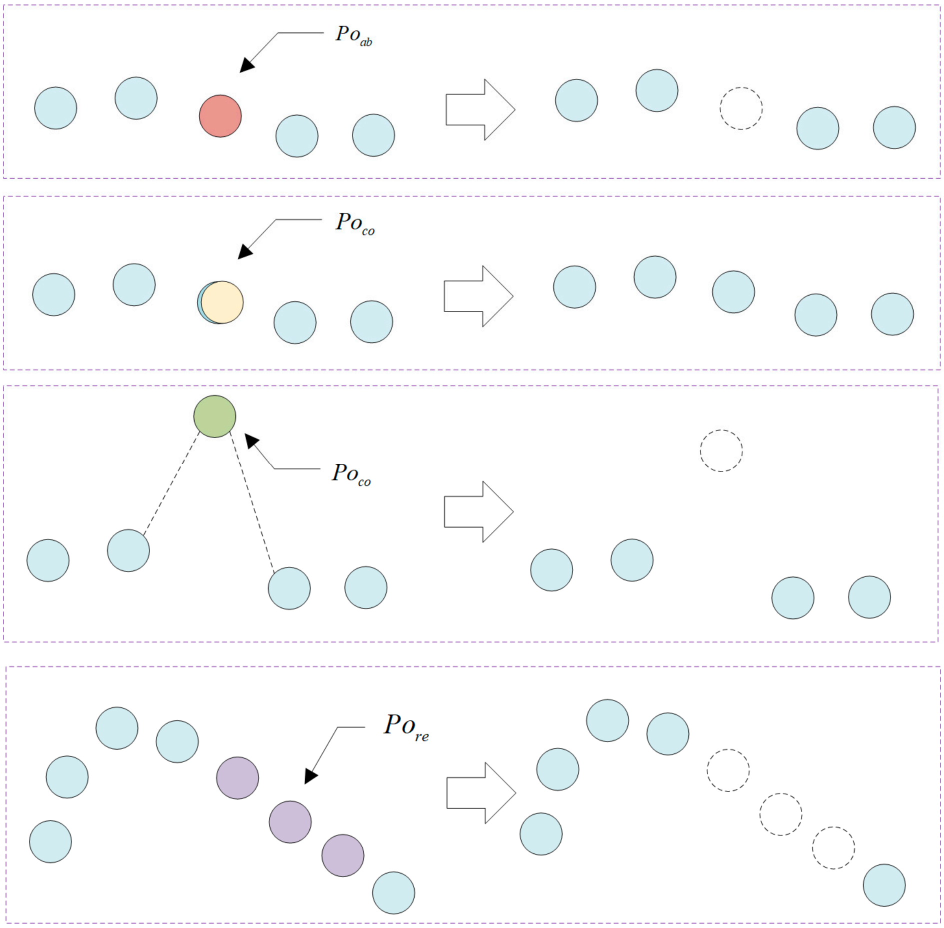

Figure 4. Removal of anomalies and redundant points.

Additionally, this TRFM model employs multi-head attention, which involves splitting the original temporal point features into multiple heads. Each head corresponds to a specific feature portion and undergoes a Self-Attention mechanism, shown in Equation (24). This calculates the attention weights for each head, which are then used to fuse the features of the current temporal point with those of other temporal points corresponding to that head. The multiple heads are then merged, as shown in Equation (25), and fed into the Feed Forward layer; the structure of the multi-head attention layer is shown in Figure 5. The Feed Forward layer essentially functions as a fully connected layer, further enhancing non-linearity and high-dimensional feature representation.

$$head_i = Self - Attention(Q_i, K_i, V_i), i = 1,2,3\ldots\mathrm{n} \tag{24}$$

$$MultHead(Q, K, V) = Concat(head_1, \ldots, head_n) \tag{25}$$

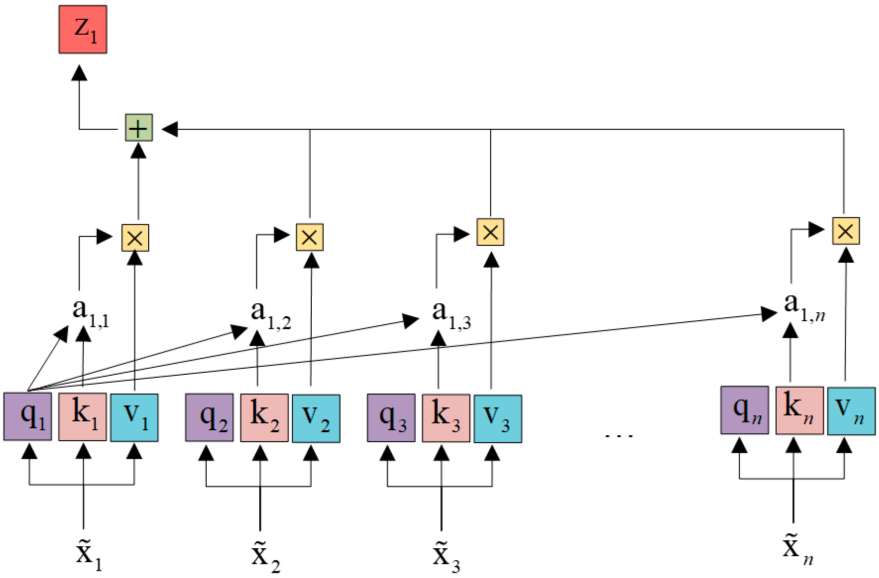

Figure 5. The Self-Attention calculation structure.

Residual networks are introduced outside the multi-head attention layer and Feed Forward layer to prevent the occurrence of gradient explosions when stacking deep networks. The outputs of both the multi-head attention and Feed Forward layers are subject to layer normalization, ensuring that the learned feature values are distributed within a certain range.

3.3.2. Encoder

The encoder primarily encodes the input temporal features. The input temporal features first pass through an encoding layer, where the value encoding of temporal point features is performed (as per Equation (26)). This is then combined with the encoding of temporal positions to form the input features of the encoder (as per Equations (27) and (28)). At this point, the input temporal point features contain both their own high-dimensional information and temporal position information.

$$VE = Relu(XW + B) \tag{26}$$

Here, VE represents value encoding, X represents the input temporal sequence, W represents the encoding weight matrix, and B represents the bias matrix. The encoding results are activated using the $Relu$ function, introducing non-linearity.

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d}}\right) \tag{27}$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{28}$$

PE represents position encoding, where pos denotes the position of the current temporal point in the input sequence. $2i$ and $2i + 1$, respectively, represent the positions of odd and even features, and d represents the feature dimension of the sequence points.

The encoder is composed of multiple stacked TRFM units. The input features of the encoder are replicated three times and multiplied by the corresponding trainable weight matrices W_Q, W_K, and W_V to generate the Q, K, and V matrices, which are then fed into the TRFM unit layer. The output dimensions of the temporal features from the multi-layer TRFM units remain consistent with those of the input. The output of the encoder is the temporal features.

3.3.3. Decoder

Similar to the encoder, the core structure of the decoder is the TRFM unit. To better generate the predicted sequence, a portion of the input sequence is used as prior information to guide the decoding. That is, the input sequence of the decoder consists of the latter part of the model's input sequence and a portion of the initialized normal distribution input sequence. This is then combined with the encoding of temporal positions to form the input of the decoder. The overall length of the input sequence of the decoder is consistent with the output sequence length of the encoder.

Notably, the TRFM unit in the decoder uses a cross-multi-head attention mechanism. The features of the output sequence from the encoder are replicated twice to form the K and V matrices. The input of the decoder is only used to compute the Q matrix. As a result, the output of the TRFM layer in the decoder also yields a sequence feature of the same length as the input. By repetitively stacking the TRFM layers of the encoder, the output undergoes a fully connected layer, and the resulting output is the final prediction of the model.

The output of the trajectory prediction model is a sequence of differential latitude and longitude values. To obtain the true predicted trajectory coordinates, they need to be calculated using the absolute regression of latitude and longitude values. Thus, the pseudo-code for the trajectory prediction process in this paper is provided in Algorithm 2.

3.3.4. Normalization

In this paper, Min–Max Normalization is used for various dimensions of the model input data. This normalization method is applied to normalize four trajectory point features: the course over ground (cog_i), speed over ground (sog_i), relative longitude (Δlon_i), and relative latitude (Δlat_i). The specific formula is as follows:

$$X^* = \frac{X - \min}{max - min} \tag{29}$$

Here, max represents the maximum value in the sample data, min represents the minimum value in the sample data, X is the original data, and X^* is the normalized data.

3.3.5. Loss Function

In this paper, $RMSE$ (Root Mean Square Error) is used as the loss function in the prediction model. $RMSE$ measures the similarity between predicted points and target points. The loss function is defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y^j - \hat{Y^j}\right| \tag{30}$$

Since the model's predictions are relative values, the output y is scaled up by a factor of 10^4 to aid convergence.

Algorithm 2: Ship Trajectory Prediction Algorithm

Input: $Traj$: Represents input trajectory data. l_{min}: Represents the min length of trajectory. T_{interp}: Represents the interpolation time interval. T_{samp}: Represents the resample time interval.
Output: Predicted trajectory : $Prediction\_Traj$.
Initialization: Nav_Traj: Used to store the trajectory of ship navigation.
Function: $Repair(\cdot)$: Used to store ship trajectories during the processing.
$Resample(\cdot)$: Used to store ship trajectories during the processing.
Process:

```
  // Step 1 Trajectory Data Preprocessing
$Tmp\_Traj \leftarrow$ Create an empty set
For $Po_i$ in $Traj$ do
  Add $Po_i$ to $Tmp\_Traj$
  If current $Po_i$ is stop point then
    If the length of $Tmp\_Traj \geq l_{min}$ do
      Add $Tmp\_Traj$ to $Nav\_Traj$
    Clear the $Tmp\_Traj$
End For
$Input\_Traj \leftarrow$ Select the last trajectory from $Nav\_Traj$
For $Po_i$ in $Input\_Traj$ do
  If $Po_i$ is removal points do
    remove $Po_i$ from $Input\_Traj$
End For
$Input\_Traj \leftarrow Repair\left(Input\_Traj,\ T_{interp}\right)$
$Input\_Traj \leftarrow Resample\left(Input\_Traj,\ T_{sample}\right)$
For $Po_i$ in $Input\_Traj$ do
  $\Delta lon_i,\ \Delta lat_i \leftarrow$ Calculate the longitude and latitude differences between $Po_i$ and $Po_{i-1}$
  $cog_i,\ sog_i \leftarrow$ Extract the course over ground and speed over ground from $Po_i$
  Add $[cog_i, sog_i, \Delta lon_i, \Delta lat_i]$ to $X$
End For
$Z \leftarrow$ Select the longitude and latitude coordinates $[lon_{end}, lat_{end}]$ of the last point of $Input_{Traj}$
// Step 2 Trajectory Prediction
$Y \leftarrow Prediction\_Model(X)$
$Current\_Position \leftarrow Z$
For $y_i$ in $Y$ do
  $Current\_Position \leftarrow Current\_Position - y_i$
  Add $Current\_Position$ to $Prediction\_Traj$
End For
```

3.4. Evaluation Metrics

In the trajectory prediction task of this paper, trajectory similarity is used to evaluate the results of data preprocessing. The evaluation metrics used include *MSE* (Mean Squared Error), *RMSE*, Average Displacement Error (*ADE*), and Final Displacement Error (*FDE*).

MSE is calculated as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y^j - \hat{Y}^j\right)^2 \quad (31)$$

RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y^j - \hat{Y}^j\right)^2} \quad (32)$$

MSE and *RMSE* are global measures of prediction accuracy, where $\hat{Y}^j$ represents the predicted relative values and Y^j represents the true relative values.

ADE is calculated as follows:

$$ADE = \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\left|Po_i^j - \hat{Po}_i^j\right| \tag{33}$$

FDE is calculated as follows:

$$FDE = \frac{1}{M}\sum_{j=1}^{M}\left|Po_{last}^j - \hat{Po}_{last}^j\right| \tag{34}$$

In the calculation formulas for *ADE* and *FDE*, *M* represents the total number of predictions, and *N* is the output step.

4. Experiments and Results

In this section, the performance of the proposed ship trajectory prediction method is validated through experiments on real AIS trajectory data. Several different evaluation metrics are applied to assess the model performance, and comparisons are made with other state-of-the-art research methods.

4.1. Trajectory Process

Details of the data preprocessing methods are explained in Section 3.2 Data Processing. The data primarily consist of one day's AIS data from the Yangtze River Basin.

This waterway experiences high ship traffic, complex river bends, and diverse ship types, making it suitable for validating ship trajectory prediction models. The collected data are stored in a database, and AIS data information is extracted, including MMSI, SOG, COG, LON, LAT, NS, and timestamps, which serve as the initial input information for data preprocessing. During data preprocessing, trajectories are segmented based on different MMSI and navigation statuses. Trajectory sequences shorter than 60 are excluded to ensure an adequate sequence length for the model input. Segmented trajectories undergo removal of outliers and redundant points, filling missing data points, and finally, uniform time interval resampling, with a set interval of 30 s. The data processing results for each step are shown in Figure 6 below.

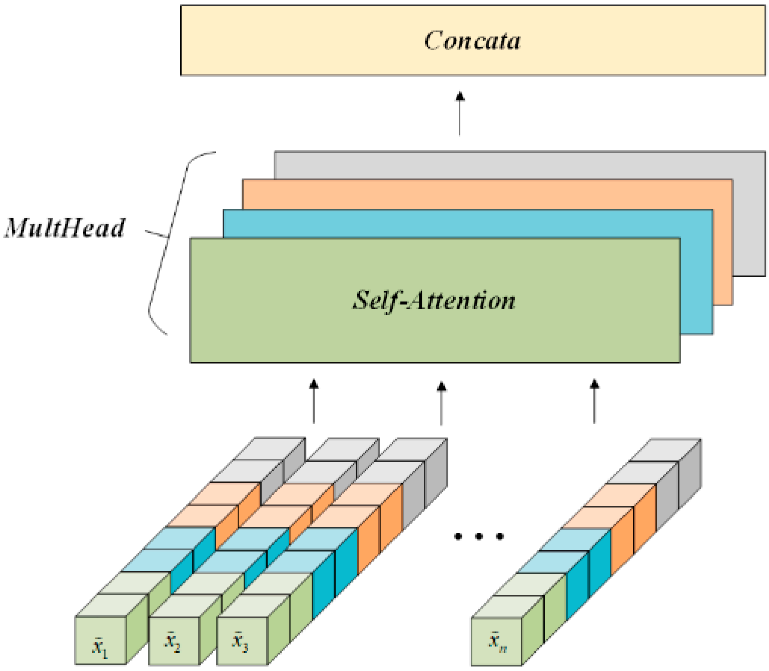

Figure 6. The multi-head attention layer.

After segmenting the trajectories, a complete navigation trajectory is obtained. After removing outliers and redundant points, the trajectory is compressed, retaining trajectory points containing changes in motion states. Missing data points are interpolated to enhance trajectory resolution. Finally, uniform time interval resampling is performed to maintain equal time intervals between trajectory points. The processed trajectories fit well with the original trajectories.

The AIS data used in this study consist of a total of 6,057,648 AIS records, with 1578 valid ship trajectories. Relative data are used as model input, using 61 historical trajectory points to predict the latitude and longitude positions of future trajectory points. After differencing to obtain relative values, the input sequence length is set to an integer of 60. In the constructed dataset, there are 40,556 training datasets, 10,140 validation datasets, and 12,674 test datasets.

4.2. Trajectory Prediction

To validate the effectiveness of the proposed TRFM model for trajectory prediction, experiments are conducted on the constructed AIS ship trajectory prediction dataset in this study. The model's performance is compared with state-of-the-art trajectory prediction models, including the LSTM model [26], ATT-LSTM model [7], CNN-LSTM model [27], and Bi-LSTM model [28]. Additionally, to validate the advantage of TRFM's encoder–decoder (ENC-DEC) structure, a comparison is made with a model using only the decoder (DEC) structure, denoted as TRFM(DEC).

The experimental environment parameters are shown in Table 2, and the model parameters for TRFM(ENC-DEC) are presented in Table 3. In the TRFM model parameters, "head" indicates the number of attention heads, "d_model" denotes the sequence feature dimensionality in the Transformer, "enc_layer" and "dec_layer" indicate the number of stacked Transformer units in the encoder and decoder, respectively, "dff" represents the feature dimensionality in the Feed Forward layer, and "label_len" is the length of prior trajectory input to the decoder.

Table 2. Experimental Environment Parameters.

Environment	Configuration
Operating System	Ubuntu 20.0
RAM	32 G
Python	3.8
Tensorflow	2.15
GPU	RTX3080 (12 G RAM)
Cpu	i7-11700k

Table 3. TRFM (ENC-DEC) Model Parameters.

Parameter	Value
head	8
d_model	512
enc_layer	2
dec_layer	2
dff	2048
label_len	15

MAE is used as the training loss during the training process. Due to the use of latitude and longitude relative values for prediction, the output prediction values are scaled up by a factor of 10,000 to enhance model convergence. The initial learning rate is set to 0.0001, and optimization is performed using the Adam stochastic gradient descent algorithm. The training process consists of 100 epochs, with a batch size of 128, totaling 40,000 training steps. Figure 7 shows the training loss curves for different models. From the figure, it can be observed that the TRFM model converges faster and achieves a lower loss compared to other models. Within the TRFM model, the ENC-DEC structure exhibits a lower loss than using the DEC structure alone, indicating superior performance.

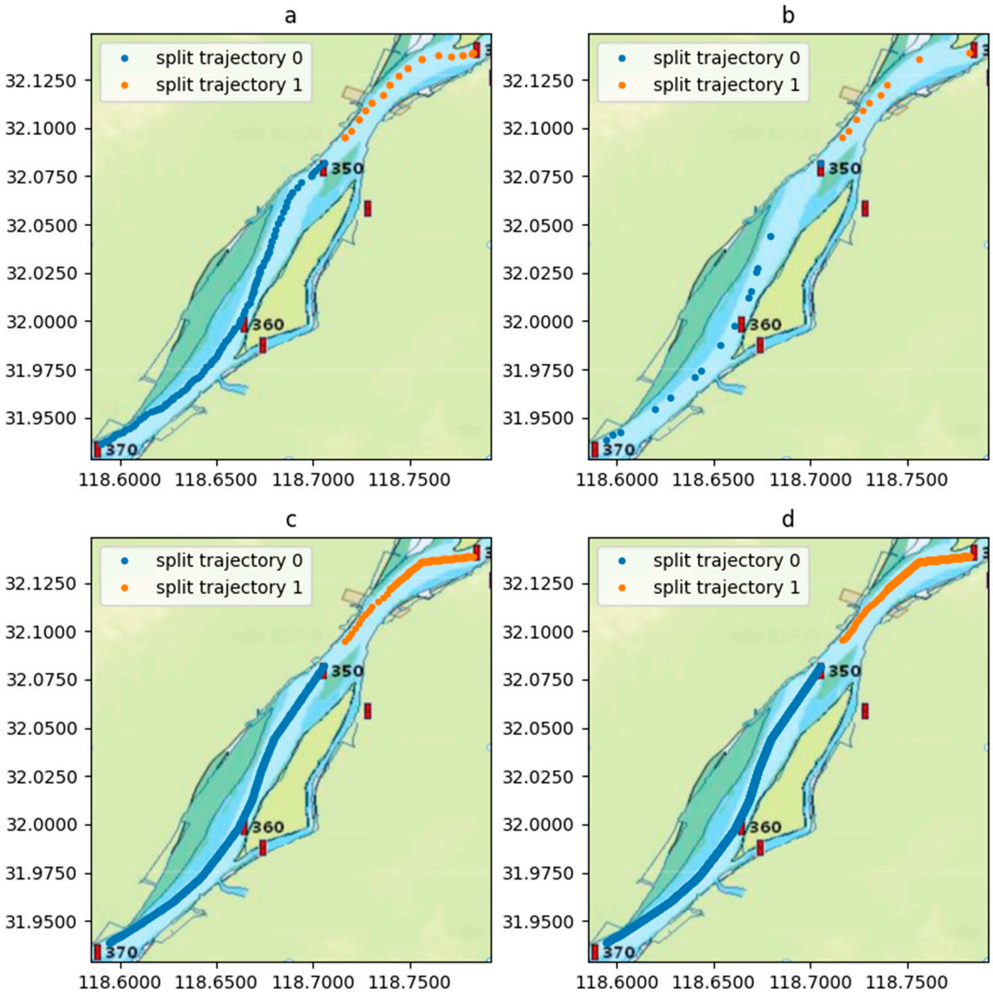

Figure 7. The data processing effects at each step: (**a**) Trajectories after segmentation. (**b**) Trajectories after segmentation. (**c**) Trajectories after segmentation. (**d**) Trajectories after uniform time interval resampling.

To measure the error between the model's predictions and the actual values, *MSE* and *RMSE* are used. *MSE* represents the error as the squared difference between predicted and actual values. A smaller *MSE* indicates better predictive performance. *RMSE*, on the other hand, also measures the error between predictions and actual values but on a smaller scale, making it more sensitive to outliers in the data. Figure 8 displays box plots showing the statistical distribution of prediction errors for different models using *MSE* and *RMSE*. From the figure, it is evident that the ENC-DEC-structured TRFM model has the smallest trajectory prediction errors compared to other models, indicating its superior accuracy and stability in trajectory prediction.

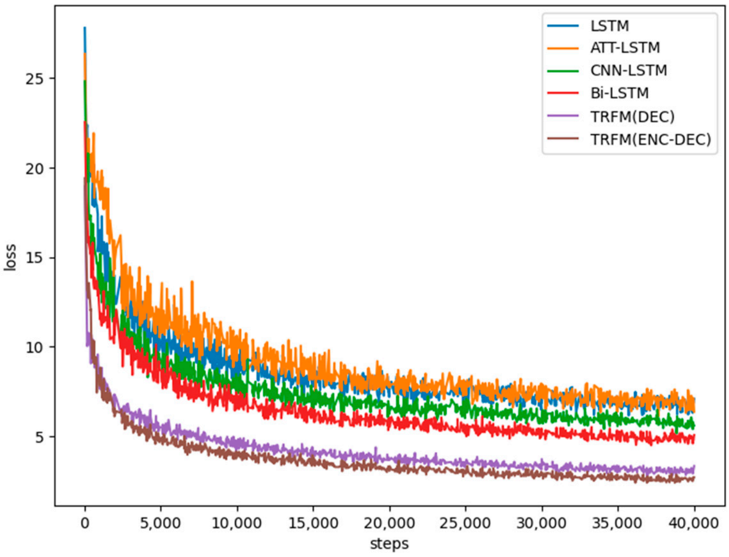

Figure 8. Training loss curves for different models.

To measure the distance error between predicted and actual trajectories, *ADE* and FDE are used to evaluate the average distance error and the endpoint distance error, respectively. *ADE* represents the global average distance error between predicted and actual trajectories, with a smaller value indicating closer alignment between predicted and actual trajectories. *FDE* represents the average error between the final predicted point and the final actual point of the trajectories, with a smaller value indicating a closer alignment of the predicted endpoint with the actual endpoint. Figure 9 presents experimental results for different models using *ADE* and *FDE* values for a prediction sequence length of 30, which corresponds to a predicted trajectory duration of 15 min. From the bar chart, it is evident that the ENC-DEC-structured TRFM model has the lowest *ADE* and *FDE* values, indicating that the TRFM model's predicted trajectories closely match the actual trajectories.

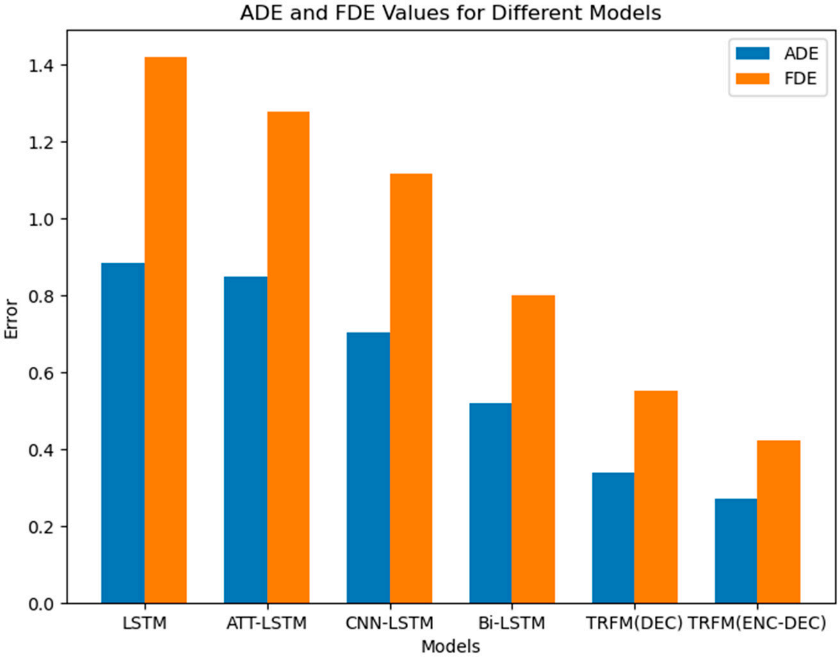

Figure 9. Bar charts of *ADE* and *FDE* for different models.

In real trajectory prediction scenarios, we compared the trajectory prediction method based on the TRFM model proposed in this paper with other trajectory prediction models and created comparison plots of predicted and actual trajectories. Figure 10 shows the comparison of predicted and actual trajectories for different models. It is clear from the figure that the ENC-DEC-structured TRFM model provides the best fit to the actual predicted trajectories, with the smallest deviations between predicted and actual trajectory points. This demonstrates that the trajectory prediction model based on the TRFM model proposed in this paper has superior predictive performance.

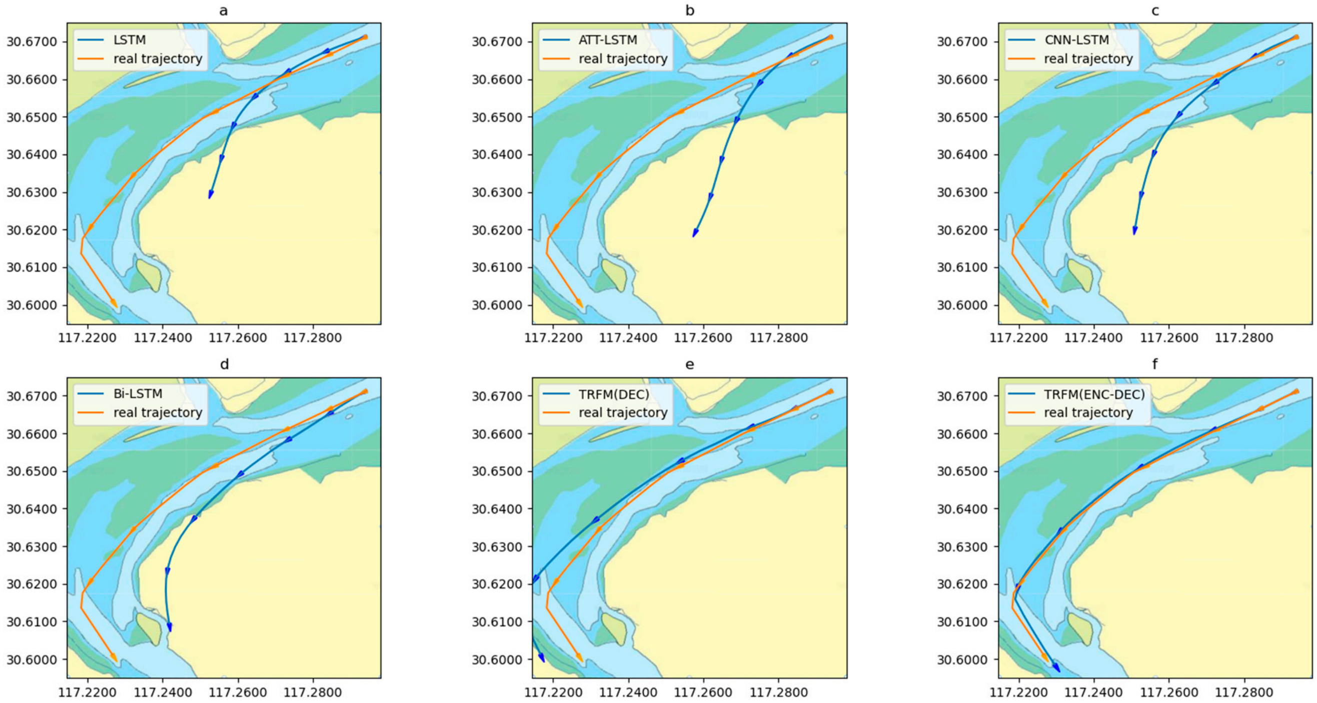

Figure 10. Comparison of predicted trajectories and actual trajectories for different models: (**a**) LSTM, (**b**) ATT-LSTM, (**c**) CNN-LSTM, (**d**) Bi-LSTM, (**e**) TRFM(DEC), (**f**) TRFM(ENC-DEC).

To further illustrate the predictive accuracy of the TRFM trajectory prediction model proposed in this paper, Table 4 presents the evaluation of average errors in longitude, latitude, and distance for different prediction sequence lengths with a time interval of 30 s. Here, 'N' represents the prediction sequence length, 'LON' represents longitude error, 'LAT' represents latitude error, and 'FDE' represents distance error. The experimental results show that the ENC-DEC-structured TRFM model has the lowest errors across different prediction sequence lengths. For a 15 min trajectory prediction, the ENC-DEC-structured TRFM model achieves an endpoint error of 0.423 km, which is a 23.51% improvement compared to using only the DEC-structured TRFM and a 47.05% improvement compared to other state-of-the-art trajectory prediction models. These results indicate that the TRFM ship trajectory prediction method proposed in this paper performs exceptionally well.

Table 4. Experimental results for different models under various prediction sequence lengths.

		LSTM	ATT-LSTM	CNN-LSTM	Bi-LSTM	TRFM (DEC)	TRFM (ENC-DEC)
N = 10 (5 min)	LON	0.00443	0.00455	0.00372	0.00282	0.00171	0.00135
	LAT	0.00410	0.00418	0.00335	0.00254	0.00153	0.00129
	FDE (km)	0.696	0.713	0.574	0.436	0.262	0.215
N = 20 (10 min)	LON	0.00725	0.00681	0.00576	0.00410	0.00275	0.00205
	LAT	0.00663	0.00616	0.00510	0.00373	0.00244	0.00201
	FDE (km)	1.126	1.053	0.881	0.634	0.420	0.331
N = 30 (15 min)	LON	0.00931	0.00816	0.00725	0.00518	0.00361	0.00265
	LAT	0.00829	0.00761	0.00652	0.00470	0.00322	0.00254
	FDE (km)	1.420	1.278	1.116	0.799	0.553	0.423

5. Conclusions

This research advances ship trajectory prediction by leveraging the comprehensive navigational data available from AIS and applying deep learning techniques. A novel preprocessing method for AIS data is introduced, aimed at reducing noise from equipment or network transmission, thus optimizing the data for a further analysis. This study centers on the implementation of the Transformer model (TRFM), which incorporates an encoder–decoder architecture and an adaptive multi-head attention mechanism. This design is proficient in extracting significant trajectory features, enhancing model inference speed, and addressing the limitations of traditional recurrent neural networks, specifically their computational inefficiency and inadequate global feature extraction capabilities. Consequently, this approach significantly improves the accuracy of trajectory predictions.

The methodology employs differential latitude and longitude values as both model inputs and outputs, facilitating the derivation of predicted trajectories through latitude and longitude regression. This technique ensures greater prediction generalization and accuracy. Experimental comparisons with leading-edge methods reveal that the proposed approach yields superior precision and reduced errors, generating predictions that closely mirror actual ship trajectories. These findings affirm the feasibility and effectiveness of the proposed algorithm, suggesting its potential to provide more accurate reference data for the navigation of future smart vessels, thereby enhancing maritime safety and operational efficiency.

Despite its innovative contributions, the trajectory prediction model based on TRFM presents certain limitations. The analysis is confined to historical AIS data, omitting external variables such as water currents and weather conditions. Predictions are limited to the geographical positions (latitude and longitude) of ships, without considering other navigational details. Moreover, the model's applicability is constrained by the requirement for a lengthy historical data series, limiting its use for datasets with shorter time series lengths.

Future research could enhance the model's comprehensiveness and accuracy by incorporating external environmental factors and extending predictions to include additional navigational parameters. Such advancements would refine the model's applicability and precision for maritime navigation forecasting.

Author Contributions: Conceptualization, Y.X.; methodology, Y.X.; validation, Y.X., T.W. and X.L.; formal analysis, Y.X.; writing—original draft preparation, X.L. and Z.Z.; writing—review and editing, Y.X. and T.W.; visualization, X.L. and Z.Z.; supervision, Y.X. and T.W.; project administration, T.W.; funding acquisition, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (grant number 2021YFB1600400), Hubei Natural Science Foundation (grant number 2021CFB324), and National Natural Science Foundation of China (grant number 52101403).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Nomenclature

AIS	Automatic Identification System
Attention–LSTM	Attention–Long Short-Term Memory
ADE	Average Displacement Error
Bi-GRU	Bi-directional Gate Recurrent Unit
COG	Course Over Ground
CRC	Change Rate of Course Over Ground
CRS	Change Rate of Speed Over Ground
FDE	Final Displacement Error
GRU	Gate Recurrent Unit
IMO	International Maritime Organization
IS-STGCNN	Improved Social Spatial–Temporal Graph Convolutional Neural Network
LSTM	Long Short-Term Memory
LON	Longitude
LAT	Latitude
MPC	Model Predictive Control
MP-LSTM	Multi-step Prediction Long Short-Term Memory
MMSI	Maritime Mobile Service Identify
MSGTIME	Message Time
MAE	Mean Absolute Error
MSE	Mean Square Error
NS	Navigational State
PESO	Parallel Encoders and a Ship-Oriented Decoder
QSD-LSTM	Quaternion Ship Domain Long Short-Term Memory
QSD	Quaternion Ship Domain
RMSE	Root Mean Square Error
SLV	Semantic Location Vector
SOG	Speed Over Ground
TDV	Trajectory Direction Vector
TRFM	Transformer
TRFM (DEC)	Transformer with Decoder
TRFM (ENC-DEC)	Transformer with Encoder and Decoder
$Traj$, $Traj_j$	The j-th Trajectory of Ship
Po_i, Po_i^j	The i-th Position in j-th Trajectory
$\mathbb{R}$	Real Number Field
$\mathbb{N}$	Positive Integer Field
cog_i	The Course Over Ground of i-th Position in Trajectory
sog_i	The Speed Over Ground of i-th Position in Trajectory
lon_i, lon_i^j	The Longitude of i-th Position in j-th Trajectory
lat_i, lat_i^j	The Latitude of i-th Position in j-th Trajectory
ns_i	The Navigation States of i-th Position in Trajectory
t_i	The Timestamp of i-th Position in Trajectory
X^j	Input Trajectory Sequence
Y^j	The Test Output Trajectory Sequence
Z^j	The Coordinates of the Last Trajectory Point of the Input Trajectory Sequence
N_K	The Number of Ship Trajectory Prediction Combinations

N_P	The Output Prediction Sequence Length
N_H	The Input History Sequence Length
$\varphi(\cdot)$	Data Processing Functions
$f_m(\cdot)$	Trajectory Prediction Function
π	Trajectory Prediction Model Parameters
θ	Trajectory Processing Function Parameters
$Traj^{mmsi}$	MMSI Trajectory Sets
$Traj^{nav}$	Navigation Trajectory Set
l_{min}	The Trajectory Length Threshold
Po_{stop}	Stopping Points
crc_i	The Change Rate of Course Over Ground for i-th Position
crs_i	The Change Rate of Speed Over Ground for i-th Position
a	A Gaussian Distribution
$b^{crs}_{min}, b^{crs}_{max}$	The Minimum and Maximum Values of the CRS Threshold Boundary
$b^{crc}_{min}, b^{crc}_{max}$	The Minimum and Maximum Values of the CRC Threshold Boundary
$\overline{x}^a, \overline{x}^a_1, \overline{x}^a_2$	The Mean of the Samples
$\sigma^a, \sigma^a_1, \sigma^a_2$	The Variance of the Samples
Po_{drop}	Removal Points
Po_{er}	Error Points
Po_{co}	Coincident Points
Po_{de}	Deviation Points
Po_{re}	Redundant Points
$dist(\cdot)$	Distance Calculation Function
V_{max}	Maximum Deviation Speed Threshold
T_{interp}	Specified Time Interval Threshold
T_{samp}	Sampling Interval
T_s, T_l	Small Time Interval, Large Time Interval
$\Delta lon_i, \Delta lon^j_i$	The Differential Values of Longitude
$\Delta lat_i, \Delta lat^j_i$	The Differential Values of Latitude
x^j_i	The Input Sequence Point
y^j_i	The Output Sequence Point
Q, K, V	The Query, Key, and Value Input
d_k	The Feature Dimension of the k-th Head.
$head_i$	The i-th Feature Head
VE	Value Encoding
X	The TRFM Unit Input Temporal Sequence
W	The Encoding Weight Matrix
B	The Bias Matrix
pos	The Position of the Current Temporal Point in the Input Sequence
d	The Feature Dimension of the Sequence Points
W_Q, W_K, W_V	The Query, Key, and Value Weight Metric
$\hat{Y}^j$	The Prediction Sequence
$\hat{Po}^j_i$	The i-th Position in j-th Prediction Trajectory
Po^j_{last}	The Last Position in j-th Trajectory
$\hat{Po}^j_{last}$	The Last Position in j-th Prediction Trajectory

References

1. Lee, E.; Mokashi, A.J.; Moon, S.Y. The maturity of automatic identification systems (AIS) and its implications for innovation. *J. Mar. Sci. Eng.* **2019**, *7*, 287. [CrossRef]
2. Last, P.; Bahlke, C.; Hering-Bertram, M.; Linsen, L. Comprehensive analysis of automatic identification system (AIS) data in regard to vessel movement prediction. *J. Navig.* **2014**, *67*, 791–809. [CrossRef]
3. Xue, J.; Wu, C.; Chen, Z. Ship AIS Data Mining and Processing Method in Bridge Waters of Inland River. In Proceedings of the 17th COTA International Conference of Transportation Professionals, Shanghai, China, 7–9 July 2017; American Society of Civil Engineers: Reston, VA, USA, 2017; pp. 4770–4779.
4. Qiao, T.; Yang, X.; Zhai, Y.; Fan, D.; Gao, K.; Bai, X.; Gao, C.; Lv, X.; Hong, H. Automatic Processing Flow Designed for Ship's AIS Signals. In Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing, Guangzhou, China, 10–12 May 2019; pp. 125–129.

5. Guo, S.; Mou, J.; Chen, L.; Chen, P. An Anomaly Detection Method for AIS Trajectory Based on Kinematic Interpolation. *J. Mar. Sci. Eng.* **2021**, *9*, 609. [CrossRef]
6. Zhao, L.; Shi, G.; Yang, J. Ship Trajectories Pre-processing Based on AIS Data. *J. Navig.* **2018**, *71*, 1210–1230. [CrossRef]
7. Liu, T.; Ma, J. Ship navigation behavior prediction based on AIS data. *IEEE Access* **2022**, *10*, 47997–48008. [CrossRef]
8. Zaman, B.; Marijan, D.; Kholodna, T. Interpolation-Based Inference of Vessel Trajectory Waypoints from Sparse AIS Data in Maritime. *J. Mar. Sci. Eng.* **2023**, *11*, 615. [CrossRef]
9. Guo, Z.; Qiang, H.; Xie, S.; Peng, X. Unsupervised knowledge discovery framework: From AIS data processing to maritime traffic networks generating. *Appl. Ocean Res.* **2024**, *146*, 103924. [CrossRef]
10. Wang, C.; Ren, H.; Li, H. Vessel trajectory prediction based on AIS data and bidirectional GRU. In Proceedings of the 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, 10–12 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 260–264.
11. Capobianco, S.; Forti, N.; Millefiori, L.M.; Braca, P.; Willett, P. Recurrent encoder-decoder networks for vessel trajectory prediction with uncertainty estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *59*, 2554–2565. [CrossRef]
12. Zhang, Y.; Han, Z.; Zhou, X.; Zhang, L.; Wang, L.; Zhen, E.; Wang, S.; Zhao, Z.; Guo, Z. PESO: A Seq2Seq-Based Vessel Trajectory Prediction Method with Parallel Encoders and Ship-Oriented Decoder. *Appl. Sci.* **2023**, *13*, 4307. [CrossRef]
13. Gao, D.; Zhu, Y.; Zhang, J.; He, Y.-K.; Yan, K.; Yan, B.-R. A novel MP-LSTM method for ship trajectory prediction based on AIS data. *Ocean Eng.* **2021**, *228*, 108956. [CrossRef]
14. Feng, H.; Cao, G.; Xu, H.; Ge, S.S. IS-STGCNN: An Improved Social spatial-temporal graph convolutional neural network for ship trajectory prediction. *Ocean Eng.* **2022**, *266*, 112960. [CrossRef]
15. Liu, X.; Yuan, H.; Xiao, C.; Wang, Y.; Yu, Q. Hybrid-driven vessel trajectory prediction based on uncertainty fusion. *Ocean Eng.* **2022**, *248*, 110836. [CrossRef]
16. Liu, R.W.; Hu, K.; Liang, M.; Li, Y.; Liu, X.; Yang, D. QSD-LSTM: Vessel trajectory prediction using long short-term memory with quaternion ship domain. *Appl. Ocean Res.* **2023**, *136*, 103592. [CrossRef]
17. Mehri, S.; Alesheikh, A.A.; Basiri, A. A context-aware approach for vessels' trajectory prediction. *Ocean Eng.* **2023**, *282*, 114916. [CrossRef]
18. Han, P.; Zhu, M.; Zhang, H. Interaction-aware short-term marine vessel trajectory prediction with deep generative models. *IEEE Trans. Ind. Inform.* **2023**, *20*, 3188–3196. [CrossRef]
19. Chen, J.; Zhang, J.; Chen, H.; Zhao, Y.; Wang, H. A TDV attention-based BiGRU network for AIS-based vessel trajectory prediction. *iScience.* **2023**, *26*, 106383. [CrossRef]
20. Wang, S.; Li, Y.; Xing, H.; Zhang, Z. Vessel trajectory prediction based on spatio-temporal graph convolutional network for complex and crowded sea areas. *Ocean Eng.* **2024**, *298*, 117232. [CrossRef]
21. Slaughter, I.; Charla, J.L.; Siderius, M.; Lipor, J. Vessel trajectory prediction with recurrent neural networks: An evaluation of datasets, features, and architectures. *J. Ocean Eng. Sci.* **2024**. *in press*. [CrossRef]
22. Li, H.; Jiao, H.; Yang, Z. AIS Data-Driven Ship Trajectory Prediction Modelling and Analysis Based on Machine Learning and Deep Learning Methods. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *175*, 103152. [CrossRef]
23. Jiang, D.; Shi, G.; Li, N.; Ma, L.; Li, W.; Shi, J. TRFM-Ls: Transformer-Based Deep Learning Method for Vessel Trajectory Prediction. *J. Mar. Sci. Eng.* **2023**, *11*, 880. [CrossRef]
24. Wu, L.; Gao, G.; Yu, J.; Zhou, F.; Yang, Y.; Wang, T. PDD: Partitioning DAG-Topology DNNs for Streaming Tasks. *IEEE Internet Things J.* **2024**, *11*, 9258–9268. [CrossRef]
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
26. Capobianco, S.; Millefiori, L.M.; Forti, N.; Braca, P.; Willett, P. Deep learning methods for vessel trajectory prediction based on recurrent neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 4329–4346. [CrossRef]
27. Lu, B.; Lin, R.; Zou, H. A Novel CNN-LSTM Method for Ship Trajectory Prediction. In Proceedings of the 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, China, 20–22 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2431–2436.
28. Yang, C.H.; Wu, C.H.; Shao, J.C.; Wang, Y.C.; Hsieh, C.M. AIS-based intelligent vessel trajectory prediction using bi-LSTM. *IEEE Access* **2022**, *10*, 24302–24315. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.