


Article

GAT-ABiGRU Based Prediction Model for AUV Trajectory

Mingxiu Zhao ^{1,2}, Jing Zhang ^{1,2,3,*} , Qin Li ¹, Junzheng Yang ³, Estevao Siga ¹ and Tianchi Zhang ^{4,*}

¹ School of Information Science and Engineering, University of Jinan, Jinan 250022, China; 202221200940@stu.ujn.edu.cn (M.Z.)

² Shandong Provincial Key Laboratory of Network-Based Intelligent Computing, University of Jinan, Jinan 250022, China

³ School of Data Intelligence, Yantai Institute of Science and Technology, Yantai 265699, China

⁴ School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

* Correspondence: ise_zhangjing@ujn.edu.cn (J.Z.); zhangtianchi@cqjtu.edu.cn (T.Z.)

Abstract: Autonomous underwater vehicles (AUVs) are critical components of current maritime operations. However, because of the complicated marine environment, AUVs are at significant risk of being lost, and such losses significantly impact the continuity and safety of aquatic activities. This article suggests a methodology for forecasting the trajectory of lost autonomous underwater vehicles (AUVs) based on GAT-ABiGRU. Firstly, the time-series data of the AUV are transformed into a graph structure to represent the dependencies between data points. Secondly, a graph attention network is utilized to capture the spatial features of the trajectory data, while an attention-based bidirectional gated recurrent unit network learns the temporal features of the trajectory data; finally, the predicted drift trajectory is obtained. The findings show that the GAT-ABiGRU model outperforms previous trajectory prediction models, is highly accurate and robust in drift trajectory prediction, and presents a new method for forecasting the trajectory of wrecked AUVs.

Keywords: AUV; trajectory prediction; GAT-ABiGRU



Citation: Zhao, M.; Zhang, J.; Li, Q.; Yang, J.; Siga, E.; Zhang, T. GAT-ABiGRU Based Prediction Model for AUV Trajectory. *Appl. Sci.* **2024**, *14*, 4184. <https://doi.org/10.3390/app14104184>

Academic Editor: Marco Troncosi

Received: 24 April 2024

Revised: 11 May 2024

Accepted: 13 May 2024

Published: 15 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Marine operations are becoming increasingly frequent with the acceleration of global ocean exploration. Autonomous underwater vehicles (AUVs), due to their prominent features, such as lightweight, small size, and high flexibility, have become indispensable tools in areas such as ocean exploration, submarine search and rescue, and marine resource development [1]. AUVs serve a vital role in expanding marine research and engineering technology and stimulating the discovery and development of aquatic resources. Moreover, they are also highly strategic in sustaining national defense security and implementing maritime strategy [2]. However, while performing underwater missions, AUVs face various risks and challenges caused by the complex marine environment and its conditions. Occasionally, incidents of loss of control or out-of-control occur. In this context, trajectory prediction becomes a pivotal technology to prevent risks and improve the safety of AUV operations. For AUVs lost in catastrophic accidents, the study and prediction of their drift trajectories can quickly and accurately determine the search area, speeding up the rescue and recovery process, and significantly reducing the cost and time of search and rescue. Predicting the drift trajectories of lost AUVs is challenging due to the need to estimate future pathways based on past trajectory data affected by variables like AUV kinematics and the marine environment. Consequently, accurate methods must be developed to predict the trajectories of wrecked AUVs, providing reliable future position and motion information.

To address the above problems, this paper proposes a GAT-ABiGRU-based trajectory prediction model for autonomous underwater vehicles (AUVs) to improve prediction accuracy and better cope with the complex underwater environment. First, we introduce the graph attention network (GAT) as a spatial feature extraction mechanism, which effectively

solves the problem of data isolation and ensures that the interactions between data features are fully considered. With GAT, the model can capture the spatial changes of the underwater environment more comprehensively, providing richer information for subsequent trajectory prediction. Secondly, we use a bidirectional gated recurrent unit network (ABiGRU) based on an attention mechanism to improve time-series data processing. This mechanism allows the model to focus on the most critical information by weighting and merging features, thereby improving prediction accuracy. Meanwhile, the bidirectional structure of ABiGRU enables it to make full use of the pre- and post-contextual information, further enhancing the sensitivity to future state changes. Finally, we fully account for the influence of marine environmental features on AUV trajectories. By taking these features as inputs, the model can better understand the relationship between ecological factors and AUV trajectories, further improving prediction accuracy. This innovative model design provides a new and effective underwater vehicle trajectory prediction tool.

The contributions of this thesis can be summarized as follows:

- A novel GAT-ABiGRU model is proposed to predict the trajectory of lost AUVs. This model utilizes a graph attention mechanism to extract spatial features from data and assigns different weights to the results. Simultaneously, the ABiGRU model incorporates an attention mechanism based on bidirectional gated recurrent units, allowing the model to respond more sensitively and integrate dynamic changes in marine environmental factors when processing time-series data.
- The time-series data in this study are arranged into a graph structure, with each time step acting as a node and varying degrees of interaction between nodes. Using a graph neural network allows the model to prioritize nodes that significantly impact the prediction outcomes by dynamically adjusting the weights of neighboring nodes.
- Various factors affecting AUV trajectory prediction, such as the kinematic information of historical AUVs and marine environmental conditions, were considered. Not only were the latitude and longitude of historical AUV data points taken into account, but ecological features, such as ocean current speed, temperature, and salinity at the AUV's disappearance location, were also considered to enhance the diversity of the data.

The document's structure is as follows: The first part outlines the research background, the formulation of the problem, and the importance of the study. The second part reviews related work, summarizing research findings relevant to the research topic. The third part details the methodology, including the framework of the methods used in this study and the construction of the GAT-ABiGRU model. The fourth part describes the experiment and presents the results analysis, providing an exhaustive description of the datasets and evaluation metrics used and a thorough analysis and discussion of the experimental results. Finally, the fifth part concludes the document with a summary and a look towards future research directions.

2. Related Work

Many trajectory prediction methods have been invented. These methods can generally be divided into two categories: one is based on dynamical models, and the other is based on neural network models.

2.1. Dynamical Modeling Based Approach

The most common dynamics modeling approaches are Gaussian process regression and Kalman filtering. Liu et al. [3] investigated the trajectory changes of underwater robots under the influence of ocean currents and waves and proposed a method to deal with and optimize the control error using a terminal sliding mode controller. This method can effectively improve the system's stability and help it cope better with the challenges of underwater complex dynamic environments. Fossen et al. [4] proposed a method based on the extended Kalman filter, which uses the AIS system to obtain information such as the position and velocity of a ship in real-time and predicts the ship's trajectory by building a ship motion model and designing a Kalman filter. Cheng [5] proposed a real-time trajectory

planning method combining a Gaussian process and incremental optimization, which applies to the path planning of an autonomous vehicle. A Gaussian process regression model is used to predict the vehicle trajectory, and then adjustment and optimization by an incremental optimization algorithm ensures that the constraints and optimization objectives are satisfied. However, given the complexity and variability of the marine environment and its inherent uncertainty, these traditional prediction methods based on dynamic modeling often need help to predict trajectories accurately. Although effective in specific contexts, these methods still have certain limitations in the face of complex marine environments, particularly regarding prediction accuracy and response to sudden changes.

2.2. Neural Network Model-Based Approach

Unlike traditional dynamics modeling methods, neural networks have demonstrated significant advantages in adapting to complex environments, improving model generality, and optimizing prediction results. In particular, recurrent neural networks (RNNs), as a typical structure among neural networks, excel in accurately predicting future trends by analyzing time-series data. However, RNNs also face the challenge of vanishing or exploding gradients, which limits their application range and prediction accuracy to some extent [6,7]. To overcome these challenges, researchers have actively sought innovations and developed a variety of improved RNN structures. Among them, the long short-term memory network (LSTM) and the gated recurrent unit (GRU) are two of the most representative improvement schemes [8,9]. These improved RNN structures not only effectively alleviate the problem of vanishing or exploding gradients but also further enhance the stability and prediction accuracy of the model, providing more reliable and efficient tools to cope with the complex and changing real-world environment. LSTM improves the gradient problem by effectively replacing the hidden layer units in traditional RNNs through their unique memory units and gating mechanisms [10]. Gers and other researchers [11] have optimized the LSTM and added a self-resetting feature that reduces the frequency of parameter computation and improves efficiency. As a simplified version of LSTM, GRU solves the overfitting problem with fewer parameters. It captures the long-term dependencies in the time series more efficiently due to its unique update-gate and reset-gate mechanisms. Suo et al. [12] cleverly used a gated recurrent unit (GRU) model to process the time-series data, significantly improving the ship's trajectory prediction accuracy. Predicting the drift trajectory of a stranded AUV is a time-series prediction problem. At present, models such as recurrent neural networks (RNNs) [13,14], long short-term memory networks (LSTMs) [15,16], and transformers [17,18] and their derivative versions have proven to be able to handle time-series prediction problems stably and have been widely used in research in recent years. Ma [16] and Tang [19] used the LSTM model for ship trajectory prediction, which promotes the rapid development of ship autonomous navigation technology. On the other hand, Ding et al. [20] further enhanced the model's ability to capture the characteristics of time-series data by introducing a dual attention mechanism, thus significantly improving the accuracy and efficiency of trajectory prediction. Dong et al. [21] proposed an innovative deep migration learning model that combines a bidirectional long short-term memory network (BiLSTM) and an attention mechanism for predicting the remaining useful life (RUL) of rolling bearings. This model improves the accuracy of RUL prediction and provides new ideas for research in related fields. In addition, Ma and Tian [22] proposed a CNN-LSTM model specifically for four-dimensional trajectory prediction of aircraft, which demonstrated higher accuracy and reliability in predicting the real-time position of aircraft (including latitude, longitude, altitude, and time) compared to traditional methods.

The introduction of graph neural networks (GNN) marks significant progress in handling graph-structured data. Initially designed to address complex graph-related problems, such as classifying molecular structures, GNNs have found widespread applications in various fields, including recommendation systems, machine translation, and time-series analysis. Particularly in the field of data prediction, the use of models such as temporal graph convolutional networks (T-GCNs) [23] and graph attention temporal convolutional

networks (GATCNs) [24] is becoming increasingly common. Researchers like Kipf and Welling [25] have proposed GNN models that rely on node relationships to capture information within graphs, effectively processing graph-structured data. Compared to traditional deep learning methods like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), GNNs are not limited to extracting local features of nodes. They can capture the overall structural information between nodes, providing a more comprehensive perspective for processing graph data. Researchers have attempted to integrate graph neural networks with deep learning techniques to handle the node features and structural information in graph-structured data more effectively. Models such as graph convolutional networks (GCN) [26], graph LSTM [27], graph attention networks (GAT) [28], and gated graph neural networks (GGNN) [29] have emerged, which not only fully utilize node features but also effectively capture the complex relationships within the graph structure, further enhancing model performance in graph data processing.

3. Materials and Methods

Figure 1 displays the framework for this paper’s methodology. First, data processing is performed, where data cleaning, outlier removal, data interpolation, and data normalization are performed in the data processing stage. Then, the processed data are fed into a prediction model for trajectory prediction, where features are extracted using a graph attention network, and the feature output is followed by trajectory prediction using a bidirectional gated recurrent unit network (ABiGRU) based on the attention mechanism. Finally, the model is compared with existing trajectory prediction neural networks to verify its correctness and robustness.

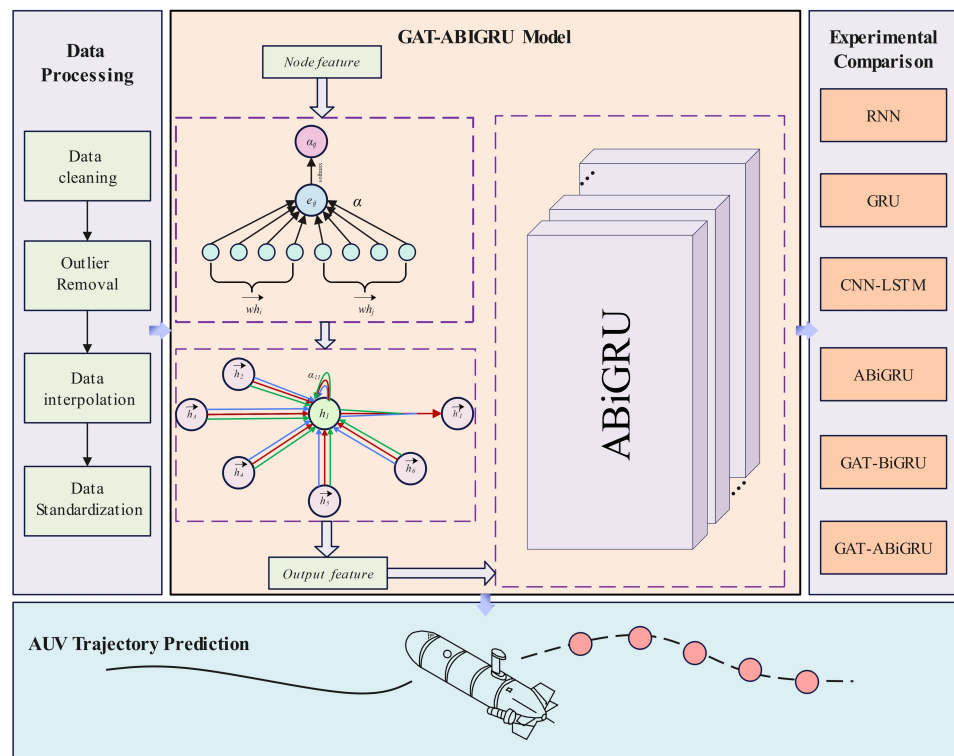


Figure 1. Framework diagram of the AUV trajectory prediction methodology.

3.1. Data Processing

The original AUV trajectory data meticulously record the kinematic characteristics of the AUV, including crucial information, such as its latitude and longitude. Before using this raw data as input for AUV trajectory prediction models, rigorous preprocessing must be undertaken to ensure the accuracy and usability of the data.

Firstly, data cleaning is crucial to address various issues in the data, such as missing values, duplicate records, or formatting errors. Through data cleaning, we can ensure the integrity and consistency of the data, providing a solid foundation for subsequent analysis. Additionally, the data may contain outliers due to measurement errors, sudden environmental changes, and other factors. These outliers can significantly interfere with the data’s authenticity and reduce subsequent analyses’ accuracy. Therefore, effective methods must be employed to identify and eliminate these outliers to ensure data reliability. During this process, the standard deviation method can be used to identify and remove outliers.

Secondly, due to equipment malfunctions, signal loss, or recording errors, there may be missing values in the historical data of the AUV. To compensate for these missing data points, we use cubic spline interpolation to perform interpolation at equal time intervals. Cubic spline interpolation can generate smooth and continuous curves based on known data points, filling in missing values and making the data more complete and constant. The formula for the spline function is as follows:

$$P(t) = \begin{cases} a_1t_1^3 + b_1t_1^2 + c_1t_1 + d_1, & \text{if } x \in [t_1, t_2] \\ a_2t_2^3 + b_2t_2^2 + c_2t_2 + d_2, & \text{if } x \in [t_2, t_3] \\ \dots \\ a_kt_k^3 + b_kt_k^2 + c_kt_k + d_k, & \text{if } x \in [t_k, t_{k+1}] \end{cases} \tag{1}$$

where $\{(t_0, P_0), (t_1, P_1) \dots (t_k, P_k)\}$ is the $(k + 1)$ th data point. Finally, the data were standardized. The formulas are as follows:

$$sta = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{2}$$

where x denotes the value before normalization, x_{min} denotes the minimum value in the data, and x_{max} denotes the maximum value in the data.

3.2. Description of the Problem

In this study, the problem of AUV trajectory prediction is viewed as a time-series prediction issue. The core of this approach lies in using historical data to predict the future state of the AUV accurately. Specifically, we use the AUV’s position information from the past T time steps to indicate the position at the next time step. Each data point of the AUV reflects its state at a specific time, and time-series analysis methods can effectively identify the correlations and underlying patterns between these data points. Through this method, we can achieve accurate predictions of the AUV trajectory.

We construct the drift trajectory network as “ G ”(t), where “ G ”(t) = (V, A(t)) denotes a set of nodes, each corresponding to the state of the recorded AUV at a particular time step t. Each time step is defined as a node, and all nodes and edges form the graph structure.

The adjacency matrix is a matrix used to describe the connection relationship between nodes in the graph structure. In our model, we use the time-varying adjacency matrix $A(t) \in \mathbb{R}^{N \times N}$ to represent the connectivity and dynamic correlation between nodes in the graph. This matrix varies over time and captures the dynamic relationships between nodes based on time-series properties and feature similarities. At any given time, if there is a connection between node i and node j , the corresponding element $A_{ij}(t)$ of the matrix is 1; if there is no connection between them, the element is 0. The formula for the adjacency matrix is as follows:

$$A_{ij}(t) = \begin{cases} 1, & \text{if node } i \text{ and } j \text{ are connected at time } t \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

In addition, to ensure that the model takes into account the relationship with other nodes and does not ignore its features when dealing with each node, we introduce the unit matrix I as a self-connecting term in the adjacency matrix to ensure that each node

takes into account its original feature information when updating its feature representation. Therefore, the modified neighborhood matrix can be expressed as:

$$\hat{A}(t) = A(t) + I \tag{4}$$

3.3. GAT

Graph attention networks (GAT), as a cutting-edge graph neural network technology, integrate neighboring node features through a unique attention mechanism by weighting them. This weighting method is entirely based on node features and does not rely on the structural information of the graph, allowing it to compute weights across time and space dimensions flexibly. For example, Wu and others [30] introduced an adaptive adjacency matrix to uncover potential spatial dependencies and combined it with extended random convolution networks to analyze long-term time trends. Additionally, Zhang and others [31] proposed an innovative graph attention LSTM model that utilizes GAT to replace traditional gating units in LSTM, which is then applied to traffic flow prediction. Meanwhile, the spatio-temporal mixed attention graph convolutional model (STMAG) [32] integrates graph convolutional networks (GCN) to capture spatial dependencies and uses a sequence-to-sequence architecture with mixed attention mechanisms to analyze temporal dependencies. The GAT network is particularly effective in processing complex spatio-temporal data because it can learn the relationships and importance between nodes. The specific network structure of GAT is illustrated in Figure 2.

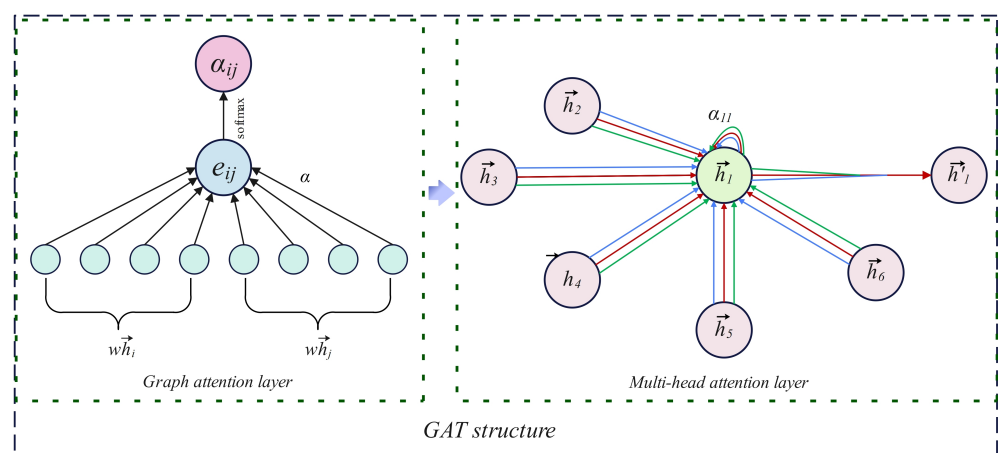


Figure 2. GAT structure.

3.4. GRU

The gated recurrent unit (GRU), introduced in 2014 [33], features simpler internal operations than LSTM. While LSTM comprises three gates, GRU consists of a reset gate (r) and an update gate (z). GRU eliminates the cell state and instead utilizes the hidden state to transmit information, resulting in fewer parameters and higher efficiency. Despite GRU's numerous simplifications compared to LSTM, it still maintains similar effectiveness. The structure of the GRU network is illustrated in Figure 3.

The specific computation process of GRU is represented as follows: At any given time, x_t represents the input at that moment, and h represents the output of the hidden state at that moment. The reset gate r_t plays a crucial role in the GRU, determining how the new input information x_t combines with the previous memory h_{t-1} . By adjusting the value of r_t , the GRU can control the degree to which it forgets previous information. When r_t approaches 0, the GRU almost wholly disregards the last memory, making the unit state similar to processing the first input in the sequence. This mechanism allows the GRU to effectively forget previous computations and reset to the initial state when necessary. The formula for the reset gate is:

$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r) \tag{5}$$

Here, σ represents the sigmoid activation function, U_r and W_r are weight matrices, x_t is the input at time t , h_{t-1} is the hidden state output from the previous time step, and b_r is the bias vector.

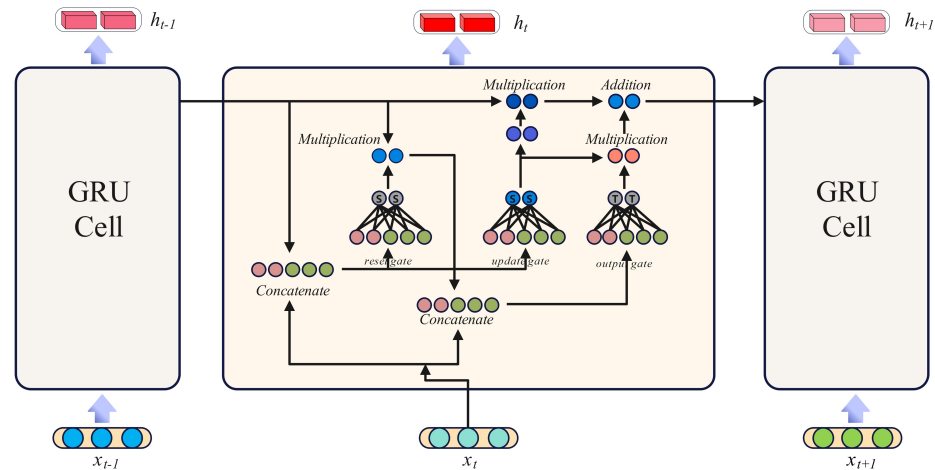


Figure 3. GRU structure.

The new hidden state \tilde{h}_t at time t is computed by integrating the current input x_t and the past hidden state h_{t-1} . This is achieved through the following formula:

$$\tilde{h}_t = \tanh(U_h x_t + W_h (r_t \circ h_{t-1})) \tag{6}$$

In this formula, U_h and W_h are weight matrices, r_t is the reset gate computed in the previous step, \circ represents element-wise multiplication, and \tanh denotes the hyperbolic tangent function.

Finally, the update gate z_t determines the rate at which the new hidden state \tilde{h}_t updates the previous hidden state h_{t-1} . Based on z_t , the current state h_t is calculated as follows:

$$z_t = \sigma(U_z x_t + W_z h_{t-1} + b_z) \tag{7}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t \tag{8}$$

Here, U_z , W_z , and b_z are weight matrices and a bias vector, respectively.

This computational procedure enables the GRU to effectively capture temporal dependencies in sequential data, adapt to new information, and selectively update its hidden state, making it well suited for tasks involving sequential data processing, such as natural language understanding and time-series prediction.

3.5. GAT-ABiRGU

Combining the spatial feature learning capabilities of the graph attention network (GAT) with the temporal analysis strengths of the gated recurrent unit (GRU), we developed a model for predicting autonomous underwater vehicle (AUV) trajectories, termed GAT-ABiGRU, as depicted in Figure 4. This model comprises four distinct layers: the input layer, the GAT layer, the ABiGRU layer, and the output layer. The GAT layer employs a multi-head attention mechanism that effectively learns the spatial features of nodes within the trajectory network, assigning tailored attention weights to each node to enhance the extraction of spatial features by focusing on the most relevant information for each node. The ABiGRU layer combines the advantages of a bidirectional GRU network with an attention mechanism, allowing for comprehensive integration of multiple dimensions of

input data. This dual approach not only captures the temporal dependencies and long-term trends in AUV trajectories but also enhances the model’s ability to accurately forecast future positions of the AUVs, producing valuable trajectory prediction results.

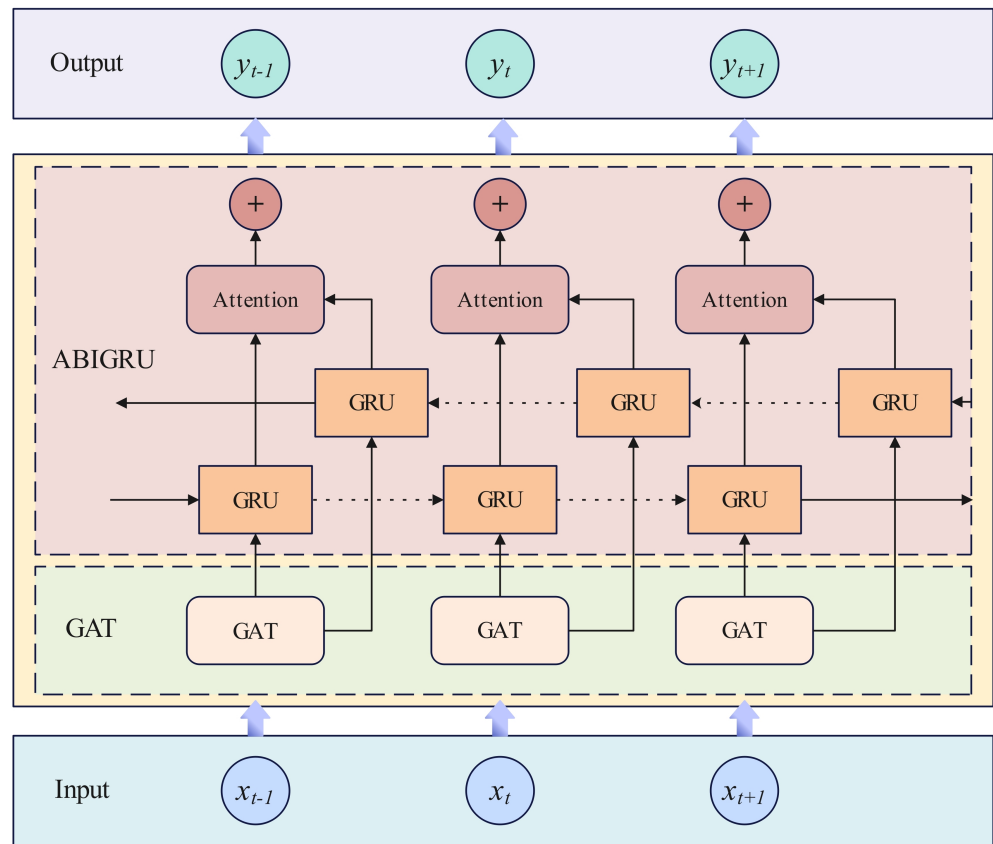


Figure 4. GAT-ABiGRU structure.

In our model, a graph attention network (GAT) layer is employed to capture the spatial features of the nodes within the trajectory network. The input to the GAT consists of a set of nodes, each represented by spatial feature vectors, denoted as $h = \{h_1, h_2, \dots, h_N\}$, $h_i \in \mathbb{R}^F$. Here, N represents the total number of nodes, and F is the dimensionality of each node’s features. The transformation weight matrix $W \in \mathbb{R}^{F' \times F}$ is utilized to enhance the dimensionality of the input features to F' .

We implement a self-attention mechanism that is shared among the edges to calculate the attention coefficients between a node and its neighbors. In the aggregation process, we assume that the central node is v_i , that the neighbor node is v_j , and set the attention coefficient from the neighbor node v_i to v_j as:

$$e_{ij} = a(W h_i, W h_j), \tag{9}$$

where $a(\cdot)$ is the function that computes the correlation between two nodes, while to simplify the computation, we restrict the calculation of this correlation to first-order neighbors.

The softmax function, combined with the leaky rectified linear unit (Leaky ReLU) activation function, is used to derive the final normalized attention coefficients. This combination ensures efficient normalization and activation, allowing for a refined adjustment of the attention mechanism’s output based on the importance of each feature in the dataset. The formula for this computation is as follows:

$$\alpha_{ij} = \text{softmax}(\text{leaky ReLU} e_{ij}) \tag{10}$$

Therefore, through the above calculations, we obtain the attention coefficients between different nodes, and the model features are updated according to the following formula:

$$h'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W h_j \right), \tag{11}$$

where W is the weight matrix of feature multiplication, α_{ij} represents the strength of attentional relations obtained from the above calculation, σ is the non-linear activation function, and $j \in N_i$ where j traversed in denotes all nodes adjacent to i . In the following description, σ represents the non-linear activation function.

As the multi-head attention mechanism (MHA) has widespread use in image processing and natural language processing tasks, it has increasingly become a focus of research attention [34]. MHA is a mechanism that amalgamates multiple self-attention structures. To address the correlations across various data levels, a multi-head attention mechanism is employed to enhance the stability of feature learning in models [35]. Specifically, we use K -independent attention mechanisms to process the above transformations, each focusing on a different level of information. The outputs of these K -independent features are then connected to form an output feature representation to combine various levels of relevant information.

Where \parallel denotes splicing and α_{ij}^K is the normalized attention coefficient under the K th attention mechanism. W^K is the corresponding weight matrix of the linear transformation under the K th attention mechanism.

The sophistication of the bidirectional recurrent neural network (BiRNN) lies in its output layer's ability to capture the complete past and future information for each time point in the input sequence [36,37]. Furthermore, the bidirectional gated recurrent unit (BiGRU), as a particular type of bidirectional RNN structure, consists of two independently functioning but collaborating components: one component is responsible for forwarding information, capturing the context from the beginning of the sequence to the current time point; while the other component is in charge of backward information transmission, gathering the context from the current time point to the end of the sequence. Through this bidirectional flow of information, BiGRU can more comprehensively understand the dependencies within sequence data, thereby enhancing the accuracy and efficiency in handling complex sequential tasks. The current state of the hidden layer of BiGRU consists of the current input x_t , the forward hidden layer state output \vec{h}_{t-1} , and the backward hidden layer state output \overleftarrow{h}_{t-1} . The BiGRU hidden layer state at time t can be obtained by the weighted sum of \vec{h}_{t-1} of the forward layer and \overleftarrow{h}_{t-1} of the hidden layer, as follows:

$$\vec{h}_t = GRU \left(x_t, \vec{h}_{t-1} \right) \tag{12}$$

$$\overleftarrow{h}_t = GRU \left(x_t, \overleftarrow{h}_{t-1} \right) \tag{13}$$

$$h_t = w_t \vec{h}_t + v_t \overleftarrow{h}_t + b_t \tag{14}$$

In the ABiGRU model, we introduce an attention mechanism to enhance BiGRU's ability to process time-series data. The ABiGRU model is shown in Figure 5. With the attention mechanism, the model can adaptively assign weights to past and future information to more refinedly capture the current prediction's essential features. Specifically, the attention mechanism evaluates and selects which information in the forward and backward GRU states is most important at the current time step to adapt and optimize the flow of information. In the ABiGRU architecture, the hidden layer state at each time step is no longer a simple weighted sum of the forward and backward hidden layer states. Instead, it is dynamically computed by an attention-weighting function that considers the forward

and backward states of all time steps. This allows ABiGRU to assign different levels of importance to the hidden states at each time step, thus paying more attention to the most critical information for predicting future states. The form of the weighting function is given in Equation:

$$h_t = \alpha_t \cdot \left(\vec{h}_t\right) + \beta_t \cdot \left(\overleftarrow{h}_t\right) + b_t, \tag{15}$$

where α_t and β_t are the forward and backward hidden state weighting coefficients, respectively, determined by the attention mechanism.

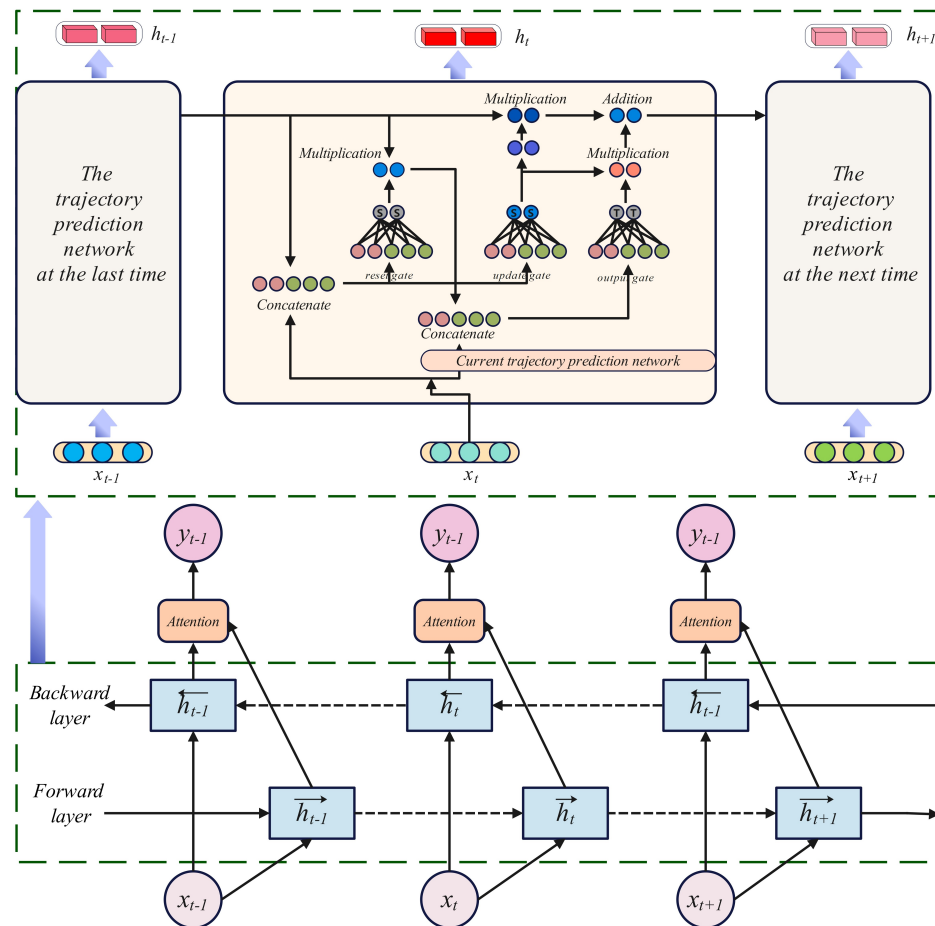


Figure 5. ABiGRU structure.

The GAT-ABiGRU model can provide accurate AUV trajectory prediction in complex marine environments by integrating the spatial analysis capability of GAT and the time-series analysis capability of ABiGRU. The model not only considers the motion characteristics of the AUV itself but also integrates a dynamic analysis of environmental factors to improve the accuracy and robustness of trajectory prediction.

4. Experiments and Results

This section describes five parts: the experimental dataset, parameter settings, evaluation indicators, analysis of prediction results, and other datasets.

4.1. Experimental Dataset

In our experimental process, a comprehensive dataset was used, which includes historical AUV kinematic features and marine environmental characteristics. This dataset consists of seven dimensions, covering crucial information, such as time, latitude, longitude, zonal ocean current speed, meridional ocean current speed, temperature, and salinity. These

valuable AUV data resources, including latitude, longitude, seawater temperature, and seawater salinity, are sourced from the BCO-DMO website [38]. The regional ocean's current environmental data were meticulously compiled and provided by the Marine Science Big Data Center. Through in-depth analysis of these multidimensional data, a more comprehensive understanding of the AUV's motion characteristics and the impact of the marine environment on its trajectory can be gained.

For this experiment, 2000 pieces of data were selected as the dataset for trajectory prediction. First, we conducted a series of preprocessing operations on the dataset, including data cleaning, outlier removal, data interpolation, and data normalization, to ensure the quality and accuracy of the data and to construct a clean and complete dataset for the experiment. Next, the processed AUV trajectory prediction dataset was logically divided, with the first 80% of the data used as the training set to train various prediction models and the remaining 20% used as the test set to evaluate the predictive performance of the models.

4.2. Parameter Settings

The study presented in this paper was carried out using the same computer configuration. CPU: Intel Core i5-9500 @ 3.00 GHz, RAM: 24.0 GB, Graphics: NVIDIA GeForce RTX 2060 (Santa Clara, CA, USA), and AUV trajectory prediction experiments in Python 3.6.

In this study, we configured some hyperparameters. The Adam optimizer is used because it performs well in sparse gradient environments, can effectively mitigate the gradient oscillation problem, and can achieve stable and efficient training convergence. Learning_rate is set to 0.001 to maintain stability during the training process and avoid instability due to excessive parameter updating. Batch_size is set to 64, and the stochastic gradient descent method is used for training to balance computational efficiency and accuracy of gradient estimation. Hidden_size is set to 32 to reduce the risk of overfitting and speed up the iteration speed. Epochs are set to 200 to ensure that the model can adequately iterate and learn from the dataset, comprehensively capture the features of the data, and progressively optimize the prediction performance. K is set to 3 to allow the model to focus on different parts of the input sequence simultaneously to improve the computational efficiency of different parts of the input sequence. In this study, we use data from seven consecutive time points as inputs to predict the latitude and longitude values of the AUV trajectory at the next instant, which fully takes into account the continuity and correlation of the time-series data and helps the model to capture the dynamic features of trajectory changes more accurately.

4.3. Evaluation Indicators

In this experiment, the mean squared error (MSE) is used as the loss function for the proposed model, quantifying the overall error between the model's predicted and actual values. To comprehensively evaluate the predictive performance of various methods, the mean absolute error (MAE) and the root mean squared error (RMSE) are also employed as evaluation metrics. These metrics reflect the deviation between predicted and actual values from different perspectives. The smaller the value, the less deviation between the predicted and actual values, thus indicating higher prediction accuracy. The formulas for these three evaluation metrics are as follows:

$$\text{MSE} = \frac{1}{\text{NUM}} \sum_t (y_i(t) - y_j(t))^2 \quad (16)$$

$$\text{RMSE} = \sqrt{\frac{1}{\text{NUM}} \sum_t (y_i(t) - y_j(t))^2} \quad (17)$$

$$\text{MAE} = \frac{1}{\text{NUM}} \sum_t |y_i(t) - y_j(t)| \quad (18)$$

where $y_i(t)$ is the predicted position of the AUV trajectory point at time t and $y_j(t)$ is the actual position of the AUV trajectory point. When the predicted trajectory is closer to the exact trajectory, the values of the error metrics, the mean square error (MSE), the mean absolute error (MAE), and the root mean square error (RMSE), will be smaller. This further indicates that the higher the model’s prediction accuracy, the better the model’s performance in predicting AUV navigation trajectories.

4.4. Analysis of Forecast Results

4.4.1. Comparative Experimental Analyses

In this experiment, we used different models to make predictions in latitude and longitude; the actual AUV trajectory is denoted as “real”, and the predicted trajectory is denoted as “prediction”. The prediction results of the GAT-ABiGRU model proposed in this paper are shown in Figure 6. Figure 6a shows the prediction results of the GAT-ABiGRU model in latitude, while Figure 6b shows the prediction results of the GAT-ABiGRU model in longitude. To make a valid comparison, in addition to the GAT-ABiGRU model proposed in this paper, we also tested several other neural network models that have performed well in the field of time-series prediction, including RNN, GRU, and CNN-LSTM. To ensure the fairness of the comparison, all models were evaluated using the same training and test sets. Figure 7a shows the prediction results of RNN, GRU, CNN-LSTM and GAT-ABiGRU models in latitude. Figure 7b shows the prediction results of these models in longitude.

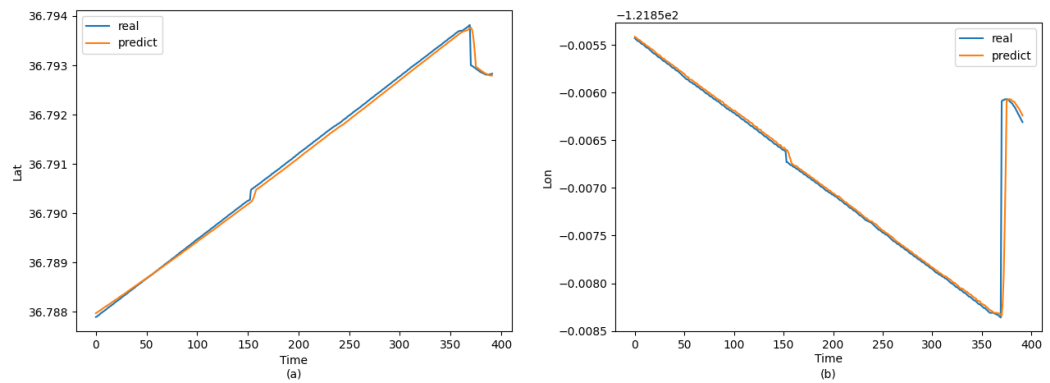


Figure 6. (a) Latitude prediction results from the GAT-ABiGRU model; (b) Longitude prediction results from the GAT-ABiGRU model.

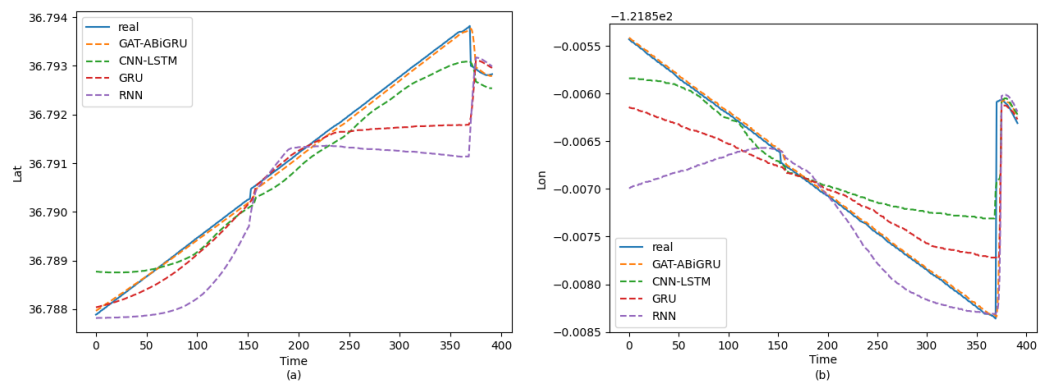


Figure 7. (a) Latitude prediction results from different models; (b) Longitude prediction results from different models.

After comparative analysis, the GAT-ABiGRU model proposed in this paper shows significant advantages in fitting real trajectories, and its predicted trajectories almost exactly match the actual trajectories with excellent performance. This is closely followed by the CNN-LSTM model, which also performs similarly excellently in predicting latitude

and longitude. In contrast, the GRU and RNN models, although capable of predicting trajectories, are relatively less accurate in matching the actual trajectory.

In Table 1, the evaluation metrics for latitude and longitude are denoted by °N and °E, respectively. The experimental results reveal that the RNN model has the lowest predictive accuracy for the AUV test dataset. It is outperformed by the GRU model and the CNN-LSTM model, with our proposed GAT-ABiGRU model demonstrating the highest performance. Compared with the GRU model, which is assessed using the RMSE metric, the GAT-ABiGRU model improves the accuracy of latitude predictions by 0.04–0.05 °N and longitude predictions by 0.03–0.04 °E. These results underscore the significant advantage of our model in accurately predicting AUV navigation trajectories.

Table 1. MSE, RMSE, and MAE value comparison among different models.

Models	Metrics	Latitude (°N)	Longitude (°E)
RNN	MSE	0.009232	0.007548
	RMSE	0.096083	0.086879
	MAE	0.135882	0.122865
GRU	MSE	0.003807	0.002014
	RMSE	0.061700	0.044877
	MAE	0.087258	0.063466
CNN-LSTM	MSE	0.002324	0.001847
	RMSE	0.048207	0.042976
	MAE	0.068176	0.060778
GAT-ABiGRU	MSE	0.000163	0.000142
	RMSE	0.012767	0.011916
	MAE	0.018055	0.016852

4.4.2. Analysis of Ablation Experiments

In this experiment, we focused on validating the impact of GAT (graph attention network) and its attention mechanism on model performance. To this end, we selected three models, ABiGRU, GAT-BiGRU, and GAT-ABiGRU, and conducted detailed experimental analyses. To maintain consistency in the experiments, the same dataset of 2000 entries is selected to ensure that all models are evaluated on the identical test and validation sets. Figure 8a,b demonstrate the prediction results of the ABiGRU model for the drift trajectories in latitude and longitude, respectively. Figure 9a,b show the prediction results of the GAT-BiGRU model for the drift trajectories in latitude and longitude. Finally, Figure 10a,b present a comprehensive comparison of the prediction results of the three models.

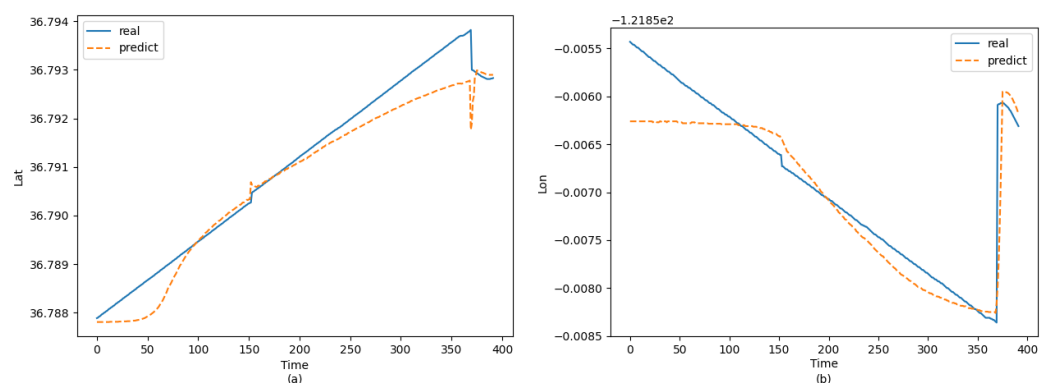


Figure 8. (a) Latitude prediction results from the ABiGRU model; (b) Longitude prediction results from the ABiGRU model.

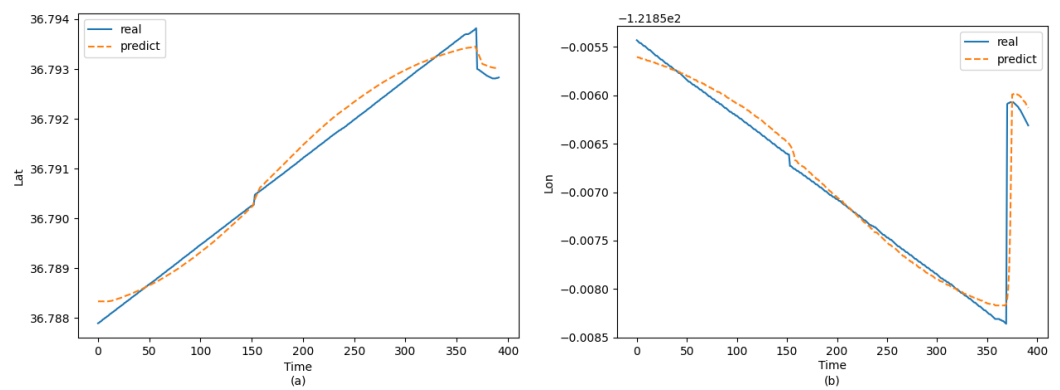


Figure 9. (a) Latitude prediction results from the GAT-BiGRU model; (b) Longitude prediction results from the GAT-BiGRU model.

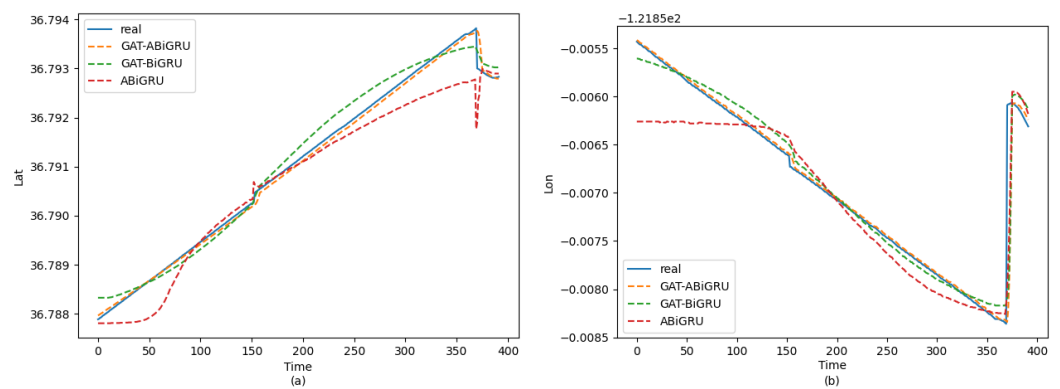


Figure 10. (a) Latitude prediction results from the GAT-ABiGRU, GAT-BiGRU, and ABiGRU models; (b) Longitude prediction results from the GAT-ABiGRU, GAT-BiGRU, and ABiGRU models.

From Figures 8–10, comparing the prediction results of latitude and longitude, it can be observed that the GAT-ABiGRU model achieves the most excellent fitting effect with the real trajectory. This model demonstrates high accuracy in predicting latitude and longitude, closely matching the real trajectory. Following it are the GAT-BiGRU model and the ABiGRU model, which although not as accurate as the GAT-ABiGRU model, still exhibit relatively outstanding performance compared to other models used in previous experiments. However, it is worth mentioning that the performance of the GAT-ABiGRU model surpasses both the GAT-BiGRU model and the ABiGRU model. This result fully demonstrates the effectiveness of the GAT network and attention mechanism in enhancing the model's performance. By introducing the GAT network and attention mechanism, the GAT-ABiGRU model can better capture the key information in the input.

From the data in Table 2, we can observe a significant improvement in the trajectory prediction performance of the GAT-ABiGRU model compared to other models. Specifically, compared to the ABiGRU model, the GAT-ABiGRU model reduces 0.018601 in RMSE (root mean square error) for latitude and 0.016806 for longitude. This remarkable improvement demonstrates that introducing graph neural networks (GAT) effectively enhances the model's prediction accuracy. Furthermore, when comparing the GAT-ABiGRU model with the GAT-BiGRU model, we can see that the GAT-ABiGRU model achieves a reduction of 0.005065 in RMSE for latitude and 0.002401 for longitude. This result indicates that incorporating an attention mechanism before the bidirectional GRU (BiGRU) output enables more precise prediction results. The attention mechanism allows the model to focus more on critical information during prediction, thus enhancing accuracy. Therefore, by combining graph neural networks and attention mechanisms, the GAT-ABiGRU model significantly

improves the accuracy of trajectory prediction and demonstrates its effectiveness and superiority in complex trajectory prediction tasks.

Table 2. MSE, RMSE, and MAE value comparison among GAT-ABiGRU, GAT-BiGRU, and ABiGRU models.

Models	Metrics	Latitude (°N)	Longitude (°E)
ABiGRU	MSE	0.000984	0.000825
	RMSE	0.031368	0.028722
	MAE	0.044362	0.040620
GAT-BiGRU	MSE	0.000318	0.000205
	RMSE	0.017832	0.014317
	MAE	0.025219	0.020248
GAT-ABiGRU	MSE	0.000163	0.000142
	RMSE	0.012767	0.011916
	MAE	0.018055	0.016852

4.4.3. Overall Experiments

Figure 11 shows the comparison results of all models with the actual trajectory in terms of latitude and longitude, with each of the six methods displaying distinct predictive outcomes.

It is clear from Figure 11 that the best performer of these algorithms in terms of prediction accuracy is our proposed GAT-ABiGRU model, whose predicted trajectory is closest to the actual trajectory. This is followed by the neural network models GAT-BiGRU, ABiGRU, CNN-LSTM, GRU, and RNN.

Figure 12 and Table 3 present the specific performance of all models in terms of the evaluation metrics, such as the mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE). Our proposed GAT-ABiGRU model achieves the best performance in all evaluation metrics, with the lowest MSE, RMSE, and MAE values. This result fully demonstrates the high accuracy characteristics of the GAT-ABiGRU model.

Table 3. MSE, RMSE, and MAE value comparison among all models.

Models	Metrics	Latitude (°N)	Longitude (°E)
RNN	MSE	0.009232	0.007548
	RMSE	0.096083	0.086879
	MAE	0.135882	0.122865
GRU	MSE	0.003807	0.002014
	RMSE	0.061700	0.044877
	MAE	0.087258	0.063466
CNN-LSTM	MSE	0.002324	0.001847
	RMSE	0.048207	0.042976
	MAE	0.068176	0.060778
ABiGRU	MSE	0.000984	0.000825
	RMSE	0.031368	0.028722
	MAE	0.044362	0.040620
GAT-BiGRU	MSE	0.000318	0.000205
	RMSE	0.017832	0.014317
	MAE	0.025219	0.020248
GAT-ABiGRU	MSE	0.000163	0.000142
	RMSE	0.012767	0.011916
	MAE	0.018055	0.016852

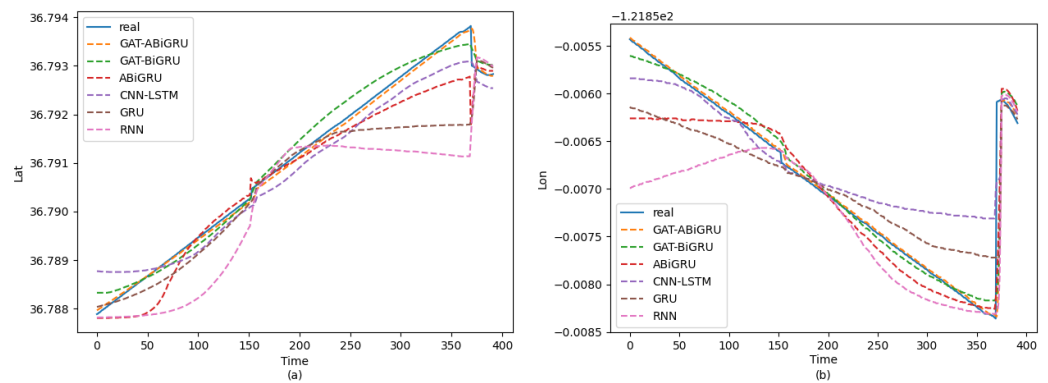


Figure 11. (a) Latitude prediction results from all models; (b) Longitude prediction results from all models.

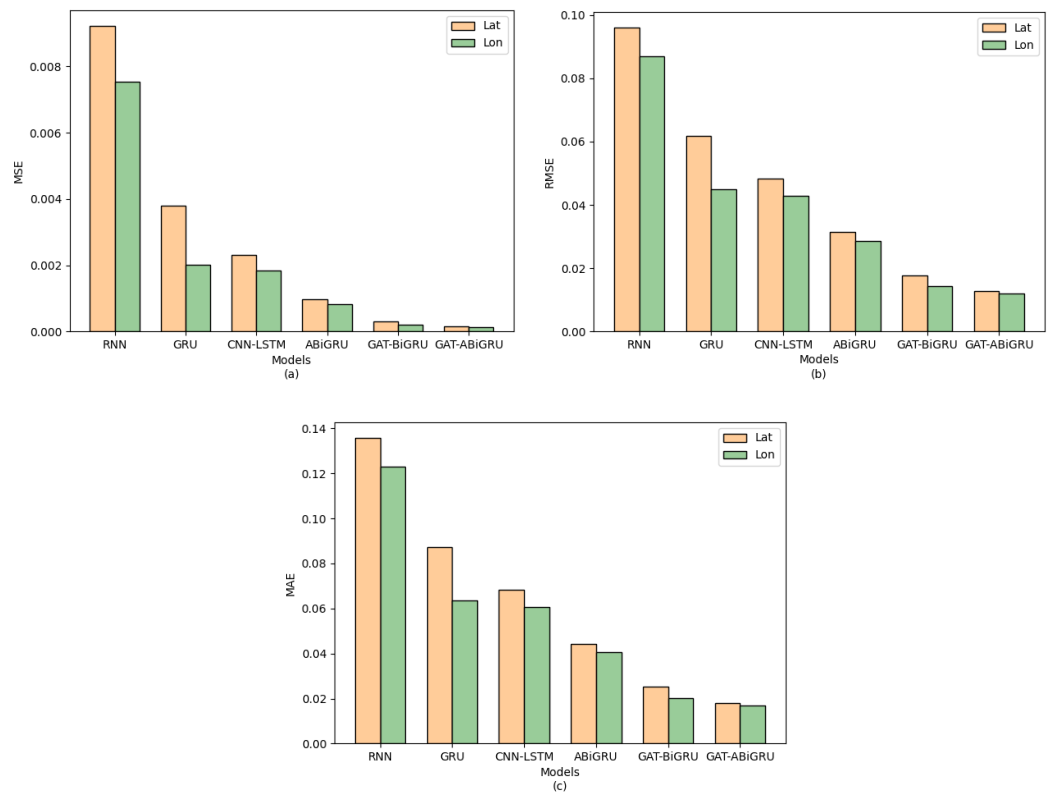


Figure 12. MSE, RMSE, and MAE value comparison among all models: (a) MSE (b) RMSE (c) MAE.

4.5. Other Datasets

To fully demonstrate the superior performance of our proposed model, we have further expanded its application testing to two time-series datasets: one for ships and another for ocean currents. These two datasets share characteristics similar to the AUV dataset, enabling our model to be effectively applied. Specifically, we selected 1700 data points from the ship dataset and 700 from the ocean current dataset. To assess the generalization ability of our model, we divided the data into training and testing sets at an 80% to 20% ratio for both datasets. Throughout all experiments, we maintained consistent model parameter settings to ensure fairness and comparability of the experimental conditions. Through this series of tests and analyses, we aim to comprehensively showcase the excellent performance of the GAT-ABiGRU model in trajectory prediction, further validating its applicability and effectiveness across different time-series datasets.

Figures 13 and 14 present the comparison results of the predicted and actual trajectories in terms of latitude and longitude for ship trajectory prediction and ocean current

trajectory prediction. These figures demonstrate high alignment between the expected and actual trajectories. The GAT-ABiGRU model exhibits excellent predictive performance across both datasets, further validating its potential in different application scenarios. Whether for ship trajectory prediction or ocean current trajectory prediction, the model demonstrates a high level of accuracy, which further reinforces its effectiveness as a tool for AUV trajectory prediction.

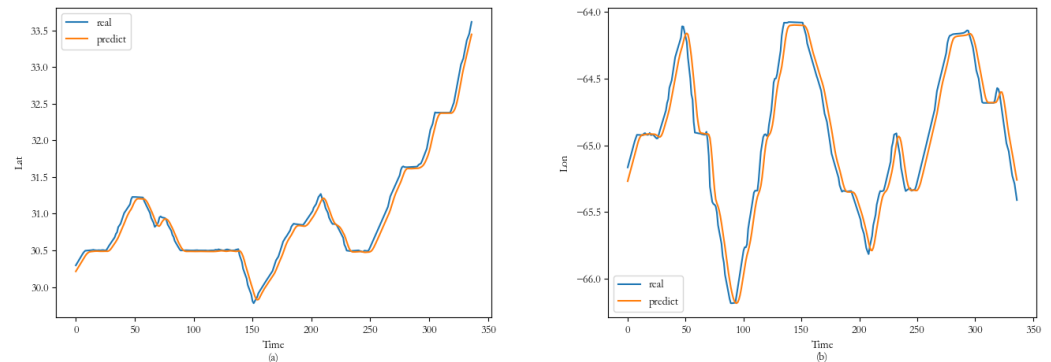


Figure 13. (a) Comparison of latitude of predicted and true ship trajectories; (b) Comparison of longitude of predicted and true ship trajectories.

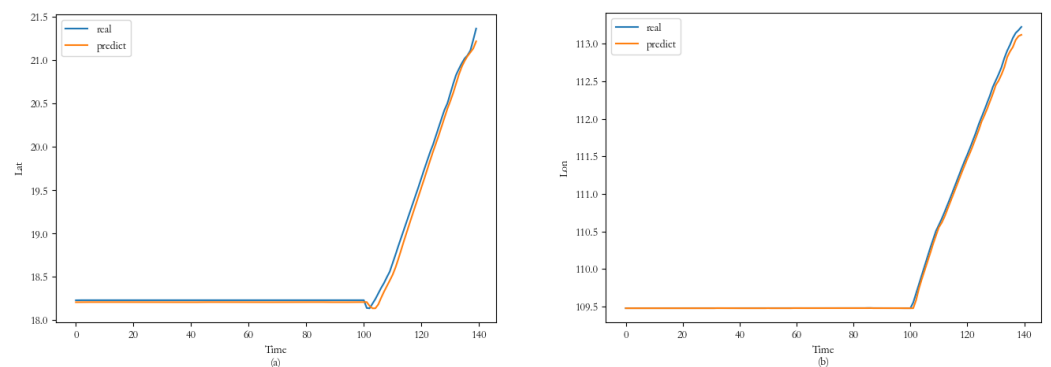


Figure 14. (a) Comparison of latitude of the predicted and true trajectories of the ocean current tracks; (b) Comparison of longitude of predicted and true trajectories of the ocean current tracks.

5. Conclusions and Future Prospects

To improve the prediction accuracy of autonomous underwater vehicle (AUV) trajectories, we propose a novel GAT-ABiGRU lost trajectory prediction model for underwater robots. The core strength of the model lies in its ability to exploit the capabilities of graph neural networks (GAT) to capture spatial dependencies in trajectory data while incorporating the time-series analysis capabilities of a bidirectional gated recurrent unit (ABiGRU) to optimize the response to the effects of changes in the complex ocean environment on AUV trajectories.

Compared to existing methods, the GAT-ABiGRU model has several significant advantages: First, it can consider both spatial and temporal characteristics of AUV trajectories, which is often neglected in traditional prediction models; second, through the graph attention mechanism, the model can automatically learn the weights between different nodes, thus capturing the key information more accurately; and finally, the time sequence analysis of the ABiGRU model optimizes the response to the impact of changes in the complex ocean environment on AUV trajectories, enabling the model to consider both forward and backward information, further improving the accuracy of the prediction. It is shown that the GAT-ABiGRU model shows a significant improvement in prediction accuracy and stability compared to existing prediction methods, especially in predicting the trajectories of lost AUVs. This is of great practical importance for the safety and efficiency of offshore

operations, as accurate trajectory prediction can help us better monitor and manage AUVs and reduce the risk of them being lost. In addition, the excellent ability of the GAT-ABiGRU model to handle complex data patterns and capture dynamic environmental changes further demonstrates its potential and value in the field of trajectory prediction. In summary, our GAT-ABiGRU model is innovative in theory and demonstrates significant application advantages in practice, providing new ideas and methods for research and application in the field of AUV trajectory prediction.

Parameter configuration is critical in ensuring model performance in neural networks. Although we have selected a set of core hyperparameters based on our rich experience and practical applications, it should be noted that different parameter configurations can significantly impact model performance. Therefore, we must continuously focus on and explore different parameter configurations and improve long-term model performance by fine-tuning and optimizing the hyperparameters to adapt to new data and application scenarios.

Future research will focus on the model's further optimization and practical application. We plan to improve further the model's ability to detect and adapt to environmental changes by expanding the size and type of dataset and integrating more ocean dynamics parameters. Meanwhile, improving the real-time computational capability will be a crucial direction for optimization to achieve fast response and real-time decision support for AUV trajectories.

Author Contributions: Conceptualization, M.Z. and J.Z.; methodology, M.Z.; software, M.Z.; validation, M.Z., T.Z. and J.Z.; formal analysis, Q.L.; investigation, Q.L. and J.Y.; resources, J.Z.; data curation, M.Z., Q.L. and J.Y.; writing—original draft preparation, M.Z.; writing—review and editing, T.Z., M.Z. and E.S.; visualization, M.Z. and J.Z.; supervision, T.Z. and J.Z.; project administration, J.Z. and T.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by 2022–2025, the National Natural Science Foundation of China under Grant No. 52171310; and 2021–2023, the National Natural Science Foundation of China under Grant (Youth) No. 52001039.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, [ise_zhangjing@ujn.edu.cn], upon reasonable request. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xiang, X.; Yu, C.; Zhang, Q. On intelligent risk analysis and critical decision of underwater robotic vehicle. *Ocean. Eng.* **2017**, *140*, 453–465. [\[CrossRef\]](#)
2. Zhu, D.; Pang, W.; Ren, K. Research status and prospect of multi-autonomous underwater vehicle cooperative search control. *J. Univ. Shanghai Sci. Technol.* **2022**, *44*, 417–428.
3. Liu, X.; Zhang, M.; Rogers, E.; Wang, Y.; Yao, F. Terminal sliding mode-based tracking control with error transformation for underwater vehicles. *Int. J. Robust Nonlinear Control* **2021**, *31*, 7186–7206. [\[CrossRef\]](#)
4. Fossen, S.; Fossen, T.I. Extended kalman filter design and motion prediction of ships using live automatic identification system (ais) data. In Proceedings of the 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 20–22 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 464–470.
5. Cheng, J.; Chen, Y.; Zhang, Q.; Gan, L.; Liu, C.; Liu, M. Real-time trajectory planning for autonomous driving with gaussian process and incremental refinement. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 8999–9005.
6. Potter, C.; Venayagamoorthy, G.K.; Kosbar, K. RNN based MIMO channel prediction. *Signal Process.* **2010**, *90*, 440–450. [\[CrossRef\]](#)
7. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)
8. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [\[CrossRef\]](#)
9. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1597–1600.

10. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
11. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [[CrossRef](#)] [[PubMed](#)]
12. Suo, Y.; Chen, W.; Claramunt, C.; Yang, S. A ship trajectory prediction framework based on a recurrent neural network. *Sensors* **2020**, *20*, 5133. [[CrossRef](#)]
13. 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](#)]
14. Kaur, K.; Kaur, P. SABDM: A self-attention based bidirectional-RNN deep model for requirements classification. *J. Softw. Evol. Process.* **2024**, *36*, e2430. [[CrossRef](#)]
15. Gülmez, B. Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Syst. Appl.* **2023**, *227*, 120346. [[CrossRef](#)]
16. Ma, H.; Zuo, Y.; Li, T. Vessel navigation behavior analysis and multiple-trajectory prediction model based on AIS data. *J. Adv. Transp.* **2022**, *2022*, 6622862. [[CrossRef](#)]
17. Geng, M.; Chen, Y.; Xia, Y.; Chen, X.M. Dynamic-learning spatial-temporal Transformer network for vehicular trajectory prediction at urban intersections. *Transp. Res. Part Emerg. Technol.* **2023**, *156*, 104330. [[CrossRef](#)]
18. Liu, R.W.; Zheng, W.; Liang, M. Spatio-temporal multi-graph transformer network for joint prediction of multiple vessel trajectories. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107625. [[CrossRef](#)]
19. Tang, H.; Yin, Y.; Shen, H. A model for vessel trajectory prediction based on long short-term memory neural network. *J. Mar. Eng. Technol.* **2022**, *21*, 136–145. [[CrossRef](#)]
20. Ding, W.; Huang, J.; Shang, G.; Wang, X.; Li, B.; Li, Y.; Liu, H. Short-term Trajectory prediction based on Hyperparametric optimisation and a dual attention mechanism. *Aerospace* **2022**, *9*, 464. [[CrossRef](#)]
21. Dong, S.; Xiao, J.; Hu, X.; Fang, N.; Liu, L.; Yao, J. Deep transfer learning based on Bi-LSTM and attention for remaining useful life prediction of rolling bearing. *Reliab. Eng. Syst. Saf.* **2023**, *230*, 108914. [[CrossRef](#)]
22. Ma, L.; Tian, S. A hybrid CNN-LSTM model for aircraft 4D trajectory prediction. *IEEE Access* **2020**, *8*, 134668–134680. [[CrossRef](#)]
23. Cai, Z.; Tan, C.; Zhang, J.; Zhu, L.; Feng, Y. DBSTGNN-Att: Dual Branch Spatio-Temporal Graph Neural Network with an Attention Mechanism for Cellular Network Traffic Prediction. *Appl. Sci.* **2024**, *14*, 2173. [[CrossRef](#)]
24. Guo, G.; Yuan, W. Short-term traffic speed forecasting based on graph attention temporal convolutional networks. *Neurocomputing* **2020**, *410*, 387–393. [[CrossRef](#)]
25. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
26. Peng, Y.; Wu, W.; Ren, J.; Yu, X. Novel GCN Model Using Dense Connection and Attention Mechanism for Text Classification. *Neural Process. Lett.* **2024**, *56*, 144. [[CrossRef](#)]
27. Kumar, S.; Mallik, A.; Panda, B. Influence maximization in social networks using transfer learning via graph-based LSTM. *Expert Syst. Appl.* **2023**, *212*, 118770. [[CrossRef](#)]
28. Wang, Y.; Yao, X.; Zhu, P.; Li, W.; Cao, M.; Hu, Q. Integrated Heterogeneous Graph Attention Network for Incomplete Multi-modal Clustering. *Int. J. Comput. Vis.* **2024**, *1–20*. [[CrossRef](#)]
29. Xu, Y.; Ji, J.; Ni, Q.; Feng, K.; Beer, M.; Chen, H. A graph-guided collaborative convolutional neural network for fault diagnosis of electromechanical systems. *Mech. Syst. Signal Process.* **2023**, *200*, 110609. [[CrossRef](#)]
30. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
31. Zhang, T.; Guo, G. Graph attention LSTM: A spatiotemporal approach for traffic flow forecasting. *IEEE Intell. Transp. Syst. Mag.* **2020**, *14*, 190–196. [[CrossRef](#)]
32. Wang, J.; Chen, Q.; Gong, H. STMAG: A spatial-temporal mixed attention graph-based convolution model for multi-data flow safety prediction. *Inf. Sci.* **2020**, *525*, 16–36. [[CrossRef](#)]
33. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
34. Nguyen, D.; Fablet, R. TrAISformer-A generative transformer for AIS trajectory prediction. *arXiv* **2021**, arXiv:2109.03958.
35. Busbridge, D.; Sherburn, D.; Cavallo, P.; Hammerla, N.Y. Relational graph attention networks. *arXiv* **2019**, arXiv:1904.05811.
36. Cho, H.; Lee, H. Biomedical named entity recognition using deep neural networks with contextual information. *BMC Bioinform.* **2019**, *20*, 735. [[CrossRef](#)] [[PubMed](#)]
37. Xian-yan, M.; Rong-yi, C.; Ya-hui, Z.; Zhenguo, Z. Multilingual short text classification based on LDA and BiLSTM-CNN neural network. In Proceedings of the Web Information Systems and Applications: 16th International Conference, WISA 2019, Qingdao, China, 20–22 September 2019; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 2019; pp. 319–323.
38. Scholin, C. *Autonomous Underwater Vehicle Monterey Bay Time Series—AUV Makai CTD. Biological and Chemical Oceanography Data Management Office (BCO-DMO). (Version 1) Version Date 15 August 2023 [If Applicable, Indicate Subset Used]*; Springer: Berlin/Heidelberg, Germany, 2023. [[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.