Journal of
Marine Science and Engineering

MDPI

Article

Autonomous Underwater Vehicle Trajectory Prediction with the Nonlinear Kepler Optimization Algorithm–Bidirectional Long Short-Term Memory–Time-Variable Attention Model

Jieen Yao [1,2], Junzheng Yang [3], Chenghao Zhang [1,2], Jing Zhang [1,2,3,*] and Tianchi Zhang [4,*]

[1] School of Information Science and Engineering, University of Jinan, Jinan 250022, China
[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) have been widely used in ocean missions. When they fail in the ocean, it is important to predict their trajectory. Existing methods rely heavily on historical trajectory data while overlooking the influence of the ocean environment on an AUV's trajectory. At the same time, these methods fail to use the dependency between variables in the trajectory. To address these challenges, this paper proposes an AUV trajectory prediction model known as the nonlinear Kepler optimization algorithm–bidirectional long short-term memory–time-variable attention (NKOA-BiLSTM-TVA) model. This paper introduces opposition-based learning during the initialization process of the KOA and improves the algorithm by incorporating a nonlinear factor into the planet position update process. We designed an attention mechanism layer that spans both time and variable dimensions, called TVA. TVA can extract features from both the time and variable dimensions of the trajectory and use the dependency between trajectory variables to predict the trajectory. First, the model uses a convolutional neural network (CNN) to extract spatial features from the trajectory. Next, it combines a BiLSTM network with TVA to predict the AUV's trajectory. Finally, the improved NKOA is used to optimize the model's hyperparameters. Experimental results show that the NKOA-BiLSTM-TVA model has an excellent parameter optimization effect and higher prediction accuracy in AUV trajectory prediction tasks. It also achieves excellent results in ship trajectory prediction.

Keywords: trajectory prediction; AUV; BiLSTM; time-variable attention (TVA); nonlinear Kepler optimization algorithm (NKOA)

check for updates

Citation: Yao, J.; Yang, J.; Zhang, C.; Zhang, J.; Zhang, T. Autonomous Underwater Vehicle Trajectory Prediction with the Nonlinear Kepler Optimization Algorithm–Bidirectional Long Short-Term Memory–Time-Variable Attention Model. *J. Mar. Sci. Eng.* **2024**, *12*, 1115. https://doi.org/10.3390/jmse12071115

Academic Editor: Weicheng Cui

Received: 10 June 2024
Revised: 28 June 2024
Accepted: 29 June 2024
Published: 2 July 2024

1. Introduction

An AUV is an autonomous underwater vehicle that is widely used in marine engineering, military, and civil fields [1,2]. Due to its importance, there are several research topics based on AUVs, including trajectory prediction, path planning, trajectory control, etc. [3,4]. The safety of AUVs is a hot research topic. It is important to predict the possible emergence or sinking points when they fail, salvage and recover AUVs, and study the causes of failure and their subsequent solutions. AUV trajectory prediction involves using mathematical models, physical models, and machine learning algorithms to predict the drift trajectory of an AUV. The goal is to predict the future trajectory based on historical trajectory data and ocean environment data. Historical trajectory data include longitude, latitude, depth, etc., while ocean environmental factors include the speed of the ocean current, oxygen concentration, salinity, etc.

AUV trajectory prediction faces many challenges. First, the ocean environment in which AUVs operate is complex and variable. These factors can affect the AUV's trajectory, making prediction more difficult. Second, sensor measurements contain errors and noise, affecting the accuracy and completeness of the data. Trajectory prediction methods are mainly divided into physics-based methods and data-driven methods [5]. Physics-based methods establish the AUV's motion equations based on Newtonian mechanics and predict its trajectory through numerical solutions. They are usually developed for specific environments and have poor adaptability. When environmental conditions change, the model needs to be readjusted or rebuilt. Additionally, accurate physical models often require complex numerical simulations, making it difficult to meet the needs of real-time prediction. Data-driven methods use deep learning methods to capture features in the trajectory data and ocean environment data for prediction. Existing deep learning methods usually rely on historical trajectory data to predict the trajectory while overlooking the impact of complex ocean environmental factors on the AUV's trajectory. Additionally, these methods often use a simple attention mechanism to extract trajectory features in the time dimension, ignoring the feature dependency between trajectory variables.

To address these challenges, this paper proposes an AUV trajectory prediction model known as NKOA-BiLSTM-TVA. First, the model uses a CNN layer to extract spatial features between trajectories and then inputs them into the BiLSTM network to capture the feature dependency at different time steps. Next, the model uses the TVA layer to extract the dependency between the time and variable dimensions in the AUV's trajectory. This allows the model to focus on the time steps and spatial locations that have a significant impact on the predicted trajectory. At the same time, the NKOA is used to optimize the model's hyperparameters. This avoids the need to manually set the model hyperparameters and improves the model's generalization ability and stability. The main contributions of this paper are as follows:

1. This paper combines AUV trajectory data with marine environment data to predict an AUV's trajectory. Our model first uses the CNN model to extract spatial features between trajectories before predicting the trajectory.

2. This study proposes a fusion attention layer across time and variable dimensions. This layer can extract the dependency between time and variable dimensions in historical trajectory data. Additionally, it can be applied to other model structures.

3. This study first initializes the KOA using opposition-based learning and integrates a nonlinear convergence factor into the algorithm to improve it. This improves the model's generalization capability and prediction accuracy.

4. This paper proposes the NKOA-BiLSTM-TVA model for AUV trajectory prediction, which combines a time-variable attention layer, a BiLSTM network, and the NKOA. This model introduces a new idea and method for AUV trajectory prediction.

The rest of this paper is structured as follows. Section 3 introduces the NKOA process and describes the proposed model; Section 4 analyzes the model's prediction results; and Section 5 presents the conclusions of this paper and possible prospects for further research.

2. Related Works

In early trajectory prediction research, physics-based methods included dynamics modeling, physical modeling, Kalman filtering, fitting polynomials, etc. [6,7]. Perera et al. [8] proposed extended Kalman filtering as an adaptive filtering algorithm for estimating the position, velocity, and acceleration in ship trajectory prediction. They achieved accurate results in predicting ship trajectories. Luo et al. [9] introduced and compared three algorithms used in ship trajectory prediction: the extended Kalman filter (EKF), least-squares support vector regression (LSSVR), and the improved LSSVR predictor.

However, due to the complex marine environment in which AUVs move, physics-based methods perform poorly. With the development of deep learning, neural network-based trajectory prediction methods have shown superior performance [10,11]. Recurrent neural networks (RNNs) have performed well in processing time-series data and are widely used in machine translation and speech recognition. At the same time, they have also been successfully applied to trajectory prediction tasks [12–15]. However, as the prediction task complexity increases, RNNs face issues such as long-term dependence, gradient vanishing, and gradient explosion. As a result, improved RNN models and attention mechanism-based models have been developed to address these problems [16]. These models have achieved better results compared to traditional RNNs in trajectory prediction tasks. Li et al. [17] designed a CNN-LSTM model for short-term AUV trajectory prediction and verified its accuracy and robustness by comparing it with RNN, gated recurrent unit (GRU), and LSTM models. Jameer et al. [18] used a deep convolutional neural network and long short-term memory (DCNN-LSTM) network with a multi-stage structure and convolutional blocks to recognize human behavior. Meanwhile, Zhu et al. [19] proposed an attention-based DCNN-LSTM method for predicting seedling transplanting time. They used a DCNN to extract spatial feature information and LSTM to capture temporal correlations in the samples. Chen et al. [20] developed a deep learning path prediction model based on BiLSTM. They achieved better performance and higher prediction accuracy compared to other network models. Jia et al. [21] proposed a 4D trajectory prediction model based on attention mechanism and LSTM. The LSTM network extracts the sequence features of the trajectory, while the attention mechanism assigns more weight to factors influencing the trajectory. Liu et al. [5] proposed an Attention-BiLSTM model to predict the drift trajectory of AUVs, combining a soft attention mechanism and BiLSTM. Yang et al. [22] proposed a multiple regression model known as CNN-BiLSTM-Attention for multi-level feature extraction of UAVs. This model uses CNN-BiLSTM as a feature extractor and enhances the model's ability to learn key information through the attention mechanism. Sun et al. [23] combined temporal pattern attention (TPA) with BiLSTM to propose a method for predicting short-term exhaust gas temperatures. They used the BiLSTM network to capture the interrelationships between different input feature variables. At the same time, they used a temporal pattern attention mechanism to capture the nonlinear dependency between different time steps and sequences.

Finding the optimal hyperparameters for a model is key to improving prediction accuracy in AUV trajectory prediction tasks. Traditional methods require numerous experiments to find the best hyperparameters, which is inefficient and can lead the model to local optima. This paper improves the Kepler optimization algorithm to optimize model hyperparameters [24]. Mokhtar et al. [25] improved the Kepler optimization algorithm to solve stochastic optimal power flow (SOPF) problems. The improved algorithm can prevent the model from falling into local optima, and the experimental results validated the effectiveness of the new algorithm. Abdel et al. [26] improved the Kepler optimization algorithm by combining a CNN with alternative image segmentation techniques. The CNN-IKOA model achieved excellent results in image segmentation compared to other models. Mohamed et al. [27] improved the Kepler optimization algorithm with a ranking-based update and exploitation improvement mechanisms. The improved algorithm reduced the possibility of the model falling into local optimal and converged to the approximate optimal solution more quickly.

Due to the lack of an attention mechanism, BiLSTM and DCNN-LSTM models cannot capture the contextual information between different trajectory points. While the CNN-BiLSTM-Attention and TPA-BiLSTM models use an attention mechanism, they overlook the feature dependencies between the trajectory variables. However, the TVA layer designed in this paper can extract features in both the time and variable dimensions. In addition, these models require extensive experiments to determine their parameters. Other KOAs only focus on improvements during the optimization process, without improving the initialization process. At the same time, these algorithms do not improve the planet position update process. This paper uses opposition-based learning during the KOA's initialization process and improves the algorithm with a nonlinear factor in the planet position update process.

3. Research Methods and Proposed Model

This section describes the methods used in our work. First, we define the AUV trajectory prediction problem and then describe the improvements to the KOA and the TVA attention layer in detail. Finally, we introduce the proposed NKOA-BiLSTM-TVA model and the process of model prediction trajectory.

3.1. Problem Definition

AUV trajectory data include latitude, longitude, and depth, which describe the precise position of the AUV in three-dimensional space. The data also include the pitch angle and roll angle, which determine the AUV's attitude. Marine environmental data include the seawater salinity, oxygen concentration, speed of the ocean currents, etc. To predict the AUV's trajectory, the trajectory data we selected include the longitude, latitude, depth, pitch angle, and roll angle. The marine environment data we selected include the seawater salinity, speed of the ocean currents, and oxygen concentration. These data are used to predict the longitude, latitude, depth, pitch angle, and roll angle of the trajectory. The trajectory data at time t can be defined as follows:

$$F(t) = \{lon_t, lat_t, alt_t, pitch_t, roll_t, salty_t, oxygen_t, speed_t\} \tag{1}$$

where $lon_t, lat_t, alt_t, pitch_t, roll_t, salty_t, oxygen_t$, and $speed_t$ represent the longitude, latitude, depth, pitch angle, and roll angle of the trajectory at time t, along with the salinity, oxygen concentration of the seawater, and speed of the ocean currents. The model's output at time $t+1$ can be described as follows:

$$O(t+1) = \{lon_{t+1}, lat_{t+1}, alt_{t+1}, pitch_{t+1}, roll_{t+1}\} = M(\{C(t-k+1), \cdots, C(t-1), C(t)\}) \tag{2}$$

where M represents our model. The model uses k historical trajectory data to predict the future position of the AUV. The model predicts only five variables ($lon, lat, alt, pitch$, and $roll$), which can determine the position and status of the AUV.

There is a temporal dependency between data points at different times in the AUV's trajectory. In addition, there is a dependency between different variables in the trajectory. Taking full advantage of the dependency between time steps and variables can lead to more accurate trajectory prediction for AUVs.

3.2. Nonlinear Kepler Optimization Algorithm

The Kepler optimization algorithm is a recently proposed optimization algorithm based on Kepler's laws of planetary motion. In this paper, we first use opposition-based learning to initialize the KOA and introduce a time-varying nonlinear convergence factor to optimize the model's hyperparameters. The process of the NKOA is described below.

Step 1: Initialization The KOA randomly initializes N planets as candidate solutions in the search space. Each planet includes d dimensions representing the problem's dimensions to be optimized. The formula for random initialization is:

$$X_{i,j} = X_{i,j}^{L} \times (1-r) + r \times X_{i,j}^{U} \tag{3}$$

where $i \in [1, N]$, $j \in [1, d]$,$X_{i,j}$ denotes the ith candidate solution. $X_{i,j}^{L}$ and $X_{i,j}^{U}$ denote the upper and lower bounds of the parameter range to be optimized for the jth parameter of the ith candidate solution, and r is a random number between 0 and 1.

Step 2: Defining Gravity The changes in the position and velocity of the planets are influenced by the gravitational force between the Sun and the planets. The gravitational force of the Sun X_{Sun} and any planet X_i can be defined as follows:

$$F_{gi}(t) = e_i \times \mu(t) \times \frac{\overline{M}_{Sun} \times \overline{m}_i}{\overline{R}^{2}_{i} + \lambda} + r_1 \tag{4}$$

where e_i is the eccentricity, M_{Sun} and m_i are the masses of the Sun and the planets, and r_1 is a random number between 0 and 1. $\overline{R_i}$ is the normalized value of the Euclidean distance $R_i(t)$ between the Sun and the planets at time t, and $\mu(t)$ is an exponential function of time t used to limit the search accuracy.

Step 3: Planetary Velocity Update The size of the planets' velocity is closely related to their distance from the Sun; when the planets are farther away from the Sun, the velocity between them is relatively small, and when the distance between them becomes smaller, the velocity between them becomes larger. According to this principle, the formula for updating the planets' velocity is as follows:

$$\overrightarrow{V_i}(t) = \begin{cases} \ell \times \left(2r_4 \overrightarrow{X}_i - \overrightarrow{X}_b\right) + \ddot{\ell} \times \left(\overrightarrow{X}_a - \overrightarrow{X}_b\right) + (1 - R_{i-norm}(t)) \times F \times \overrightarrow{U}_1 \times \overrightarrow{r_5} \times \left(\overrightarrow{X}_i^U - \overrightarrow{X}_i^L\right), if R_{i-norm}(t) \leq 0.5 \\ r_4 \times L \times \left(\overrightarrow{X}_a - \overrightarrow{X}_i\right) + (1 - R_{i-norm}(t)) \times F \times U_2 \times \overrightarrow{r_5} \times \left(r_3 \overrightarrow{X}_i^U - \overrightarrow{X}_i^L\right), Else \end{cases} \tag{5}$$

$$F = \begin{cases} 1 & if \quad r_4 \leq 0.5 \\ -1 & Else \end{cases} \tag{6}$$

where $\overrightarrow{V_i}(t)$ is the velocity of planet i at time t. r_3 and r_4 are random numbers between 0 and 1, and $\overrightarrow{r_5}$ and $\overrightarrow{r_6}$ are randomly generated vectors between 0 and 1. F represents a parametric factor that can change the direction of planetary motion. $\overrightarrow{X}_a$ and $\overrightarrow{X}_b$ denote randomly selected initial solutions within the population.

Step 4: Elimination of Local Optimal Solutions The algorithm sets a parameter factor F that changes the search direction, which can prevent the algorithm from becoming trapped in the local optimal solution. It also allows the algorithm to better search for the optimal solution in the entire search space.

Step 5: Updating the Planet's Position With the planet's velocity update formula, the planet's position update formula is as follows:

$$\overrightarrow{X}_i(t+1) = \overrightarrow{X}_i(t) + F \times \overrightarrow{V_i}(t) + \left(F_{gi}(t) + |r|\right) \times \overrightarrow{U} \times \left(\overrightarrow{X}_{Sun}(t) - \overrightarrow{X}_i(t)\right) \tag{7}$$

Step 6: Updating distance from the Sun The algorithm has two phases—exploration and exploitation (see Figure 1)—simulating the changing distance between the Sun and the planets over time. When a planet is close to the sun, the algorithm is in the exploitation phase, optimizing the exploitation operator. When a planet is far from the Sun, the algorithm is in the exploration phase, optimizing the exploration operator. This process depends on a tuning parameter h. This process can be defined as follows:

$$\overrightarrow{X}_i(t+1) = \overrightarrow{X}_i(t) \times \overrightarrow{U}_1 + \left(1 - \overrightarrow{U}_1\right) \times \left(\frac{\overrightarrow{X}_i(t) + \overrightarrow{X}_{Sun} + \overrightarrow{X}_a(t)}{3} + h \times \left(\frac{\overrightarrow{X}_i(t) + \overrightarrow{X}_{Sun} + \overrightarrow{X}_a(t)}{3} - \overrightarrow{X}_b(t)\right)\right) \tag{8}$$

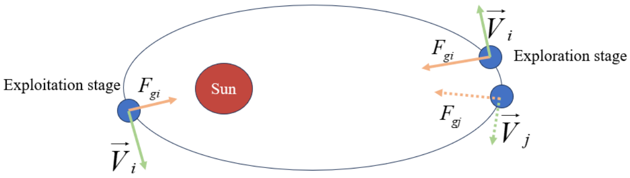


Figure 1. Two states of exploration and exploitation.

Step 7: Position Updating Policy To ensure that the Sun and planets are in optimal positions, the algorithm adopts a position-updating policy, which is defined as follows:

$$\overrightarrow{X}_{i,new}(t+1) = \begin{cases} \overrightarrow{X}_i(t+1) \quad if \quad f\left(\overrightarrow{X}_i(t+1)\right) \leq f\left(\overrightarrow{X}_i(t)\right) \\ \overrightarrow{X}_i(t) \qquad Else \end{cases} \tag{9}$$

3.2.1. Opposition-Based Learning Population Initialization

Opposition-based learning [28] is a population initialization method. For a single solution, it can generate an opposite solution in the solution space, allowing the algorithm to consider both the current solution and its opposite solution during optimization. If the opposite solution is better than the current solution, it can be chosen as a new candidate solution. By considering the initial solution and its opposite solution, the search process can cover a wider range of the solution space. This diversity helps reduce the possibility of the algorithm falling into a local optimum. The introduction of opposition-based learning enhances the algorithm's global search capability and accelerates its convergence. For the jth parameter of the ith planet, a reverse solution ${X_{i,j}}'$ can be generated based on Equation (10).

$$ {X_{i,j}}' = X_{i,j}^{L} + X_{i,j}^{U} - X_{i,j} \tag{10} $$

If the opposite solution is better than the current solution, it can be selected as a new candidate solution. This approach allows the algorithm to quickly move toward more promising search regions.

3.2.2. Nonlinear Convergence Factor

The parameter factor F in the KOA can change the search direction, preventing the algorithm from becoming trapped in the local optimal solution. However, its principle only changes the direction of a planet's movement to avoid the local optimum. To further enhance the local optimization ability of the KOA, we integrated a time-varying nonlinear convergence factor into it. This integration allows the algorithm to efficiently and accurately find the global optimum during both the exploration and exploitation phases. The formula for the nonlinear convergence factor is as follows:

$$ a = 1 - (\nu + e)\left(\frac{t}{T_{\max}}\right)^{\lambda} + \theta\varphi \tag{11} $$

where e represents Euler's constant; ν , λ , and φ are constant term coefficients; and θ is a constant term between 0 and 1. t and T denote the current iteration and the maximum iteration of the algorithm.

As seen in the formula, the nonlinear convergence factor decreases nonlinearly with the number of iterations. This can be combined with Equation (7). Thus, the nonlinear convergence factor helps the algorithm balance the breadth and depth of the search during the exploration and exploitation phases. At the beginning of the algorithm, the planets can move further, allowing the algorithm to better explore and reach the global optimal solution during the exploration phase. As the number of iterations increases, the planets move slower, allowing the algorithm to more accurately reach the local optimal solutions during the exploitation phase. The formula for updating a planet's position in the NKOA is as follows:

$$ \overrightarrow{X}_i(t+1) = \overrightarrow{X}_i(t) + F \times \overrightarrow{V_i}(t) \times \alpha + \left(F_{gi}(t) + |r|\right) \times \overrightarrow{U} \times \left(\overrightarrow{X}_{Sun}(t) - \overrightarrow{X}_i(t)\right) \tag{12} $$

We initialize the KOA's population using opposition-based learning and integrate a nonlinear convergence factor into the KOA. The NKOA uses Equations (3) and (10) to initialize the population and Equation (12) to update a planet's position.

3.3. BiLSTM Model

A RNN is a model used for processing sequence data. As shown in Figure 2, long short-term memory (LSTM) is an improvement of the RNN model with internal mechanisms, including forget gates, input gates, and output gates [29]. The forget gate handles the current input and the output of the previous hidden state; the input gate determines which information is input into the network; and the output gate determines which information from the cell state needs to be output.

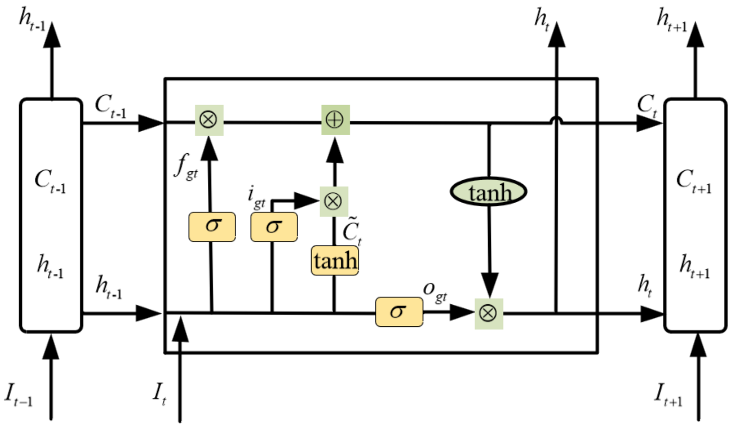


Figure 2. LSTM model structure diagram.

In Figure 2, I_t is the input data at time t, f_{gt} is the forget gate, i_{gt} is the input gate, o_{gt} is the output gate, h_t is the current hidden state, C_t is the output at the current time, and $\tilde{C}_t$ is the temporary state of the cell. Their formulas are as follows:

$$f_{gt} = \sigma\left(W_{fh} \cdot h_{t-1} + W_{fI} \cdot I_t + b_f\right) \tag{13}$$

$$i_{gt} = \sigma(W_{ih} \cdot h_{t-1} + W_{iI} \cdot I_t + b_i) \tag{14}$$

$$o_{gt} = \sigma(W_{oh} \cdot h_{t-1} + W_{oI} \cdot I_t + b_o) \tag{15}$$

$$h_t = o_{gt} * \tanh(C_t) \tag{16}$$

$$\tilde{C}_t = \tanh(W_{Ch} \cdot h_{t-1} + W_{CI} \cdot I_t + b_C) \tag{17}$$

$$C_t = f_{gt} * C_{t-1} + i_{gt} * \tilde{C}_t \tag{18}$$

where W is the weight matrix the network learns, b is the bias parameter; and $\sigma(z)$ is the sigmoid activation function. The formulas for $\sigma(z)$ and $\tanh(x)$ are as follows:

$$\sigma(z) = \frac{1}{1 - e^{-z}} \tag{19}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{20}$$

BiLSTM can use information from both past and future moments, effectively exploiting the dependency between historical and future time-step data [30–32]. Figure 3 shows the structure of the BiLSTM network.

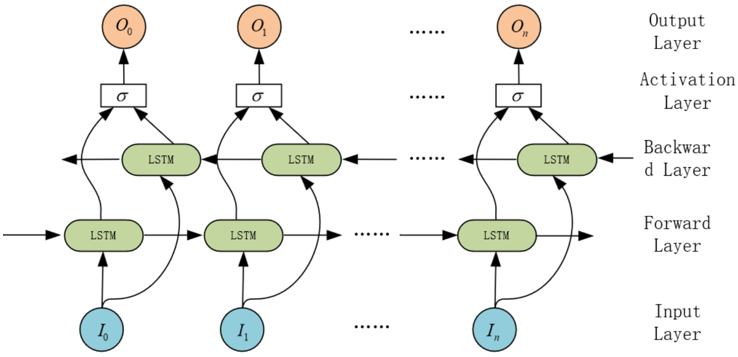


Figure 3. BiLSTM network structure.

The backward layer of BiLSTM captures the dependency on future time-step data. The forward layer allows the model to integrate historical and future time-step data for prediction. Thus, BiLSTM enhances the feature representation capability and can extract richer sequence features.

3.4. Time-Variable Attention Layer

3.4.1. Multi-Head attention

The multi-head attention mechanism learns different weight parameters by stacking the self-attention mechanism multiple times [33,34]. For the same input X, multiple sets of different parameter matrices are used to perform dot-product operations with the input to obtain different attention matrices: Q, K, and V. Then, the outputs of each attention head are concatenated and linearly transformed [35]. This process can be represented as follows:

$$MHA(Q, K, V) = Concat(Head_1 \ldots Head_n)W^o \tag{21}$$

where each $Head_i$ is a single self-attention mechanism. Figure 4 shows the model structures of the multi-head attention and self-attention.

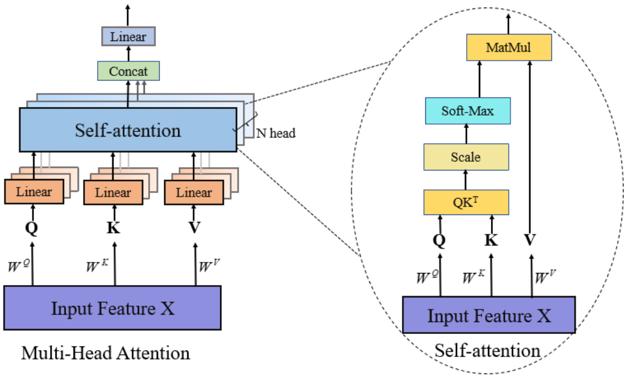


Figure 4. (**left**) Multi-head attention mechanism structure; (**right**) self-attention mechanism structure.

Existing trajectory prediction methods often use an attention mechanism to capture temporal dependency [36,37]. These methods embed data into vectors in the temporal dimension and then analyze the dependency between vectors. However, this approach does not consider the dependency among different variables and the influence of the marine environment on the trajectory. In our work, we propose a time-variable attention layer to solve these issues. Figure 5 shows the structure of the TVA layer.

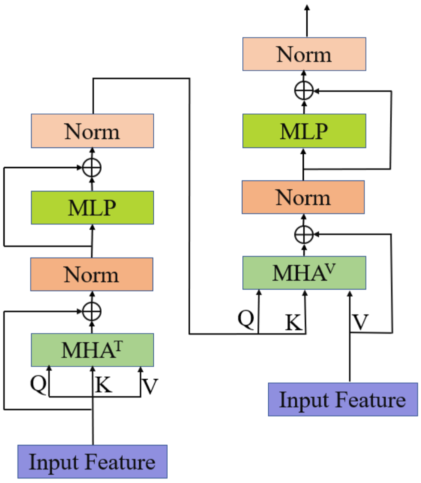


Figure 5. Time-variable attention layer structure. (**left**) Time dimension attention. (**right**) Variable dimension attention.

The TVA layer consists of a time dimension attention and a variable dimension attention. The time dimension attention mainly captures dependency between trajectories at different time steps. The variable dimension attention captures the dependency between different variables. The input data are first processed through a multi-head attention mechanism, and the output is then fed into a normalization layer. The normalization layer and the attention layer are combined through the residual connection. Then, the output is fed into the multi-layer perceptron (MLP) layer. The final result is output through a normalization layer.

3.4.2. Time Dimension Attention

In the time dimension attention stage, the input of the TVA layer is a given two-dimensional tensor $O^C \in R^{S \times D}$. Here, S represents the input time steps, and D represents the number of input trajectory variable dimensions. $O^C_{i,:}$ denotes all variable dimensions of the ith time step, where as $O^C_{:,d}$ denotes all time steps in the dth variable dimension. O^C denotes the information output from the BiLSTM hidden layer. The process of extracting the trajectory feature dependency in the time dimension through the time dimension attention can be defined as follows:

$$\widehat{O}^T = LayerNorm(MHA^T(O^C_{:,d}, O^C_{:,d}, O^C_{:,d}) + O^C_{:,d}) \tag{22}$$

$$O^T = LayerNorm(MLP(\widehat{O}^T) + \widehat{O}^T) \tag{23}$$

where $MHA^T()$ uses the multi-head attention mechanism to capture the dependency in the time dimension. O^T is the output of the time dimension attention. All variables share the attention parameter. LayerNorm performs the normalization operation, and the MLP represents the forward fully connected layer. When extracting the time dimension dependency, all variables are embedded into a feature vector to extract the dependency for each time step. Thus, the model can extract the complete dependency of trajectory data in the time dimension.

3.4.3. Variable Dimension Attention

The variable dimension attention takes the output of the time dimension attention and the original features as input. For each variable, it extracts the feature dependency across all time dimensions of a sliding window. This enables a complete extraction of the dependency between different trajectory variables. The variable dimension attention assigns more weight to trajectory variables that significantly impact the AUV's future trajectory. In addition, the marine environment can be fully used to predict the AUV's trajectory. The process of the variable dimension attention can be defined as follows:

$$\widehat{O}^V = LayerNorm(MHA^V(O^T_{i,:}, O^T_{i,:}, O^C_{i,:}) + O^C_{i,:}) \tag{24}$$

$$O^V = LayerNorm(MLP(\widehat{O}^V) + \widehat{O}^V) \tag{25}$$

where $MHA^V()$ uses multi-head attention to capture the feature dependencies in trajectory variables. O^V is the output after temporal and variable dimension dependency extraction. The process of the TVA layer can be expressed as follows:

$$Y = O^V = TVA(O^C) \tag{26}$$

where O^C and Y denote the inputs and outputs of the TVA layer.

3.5. BiLSTM-TVA

This paper combines BiLSTM with the TVA layer to construct the BiLSTM-TVA model for predicting the AUV's drift trajectory and also adds a CNN layer to the model (see Figure 6). The BiLSTM-TVA model includes the input layer, the CNN layer, the BiLSTM layer, the TVA layer, and the output layer. First, the input layer preprocesses the AUV trajectory data and creates a dataset according to the sliding window principle. Next, these data are input into the CNN layer to extract the spatial features of the AUV's trajectory. After extracting the spatial features between trajectories with the CNN network, the BiLSTM network captures the temporal dependency within the trajectory. The output of the BiLSTM hidden layer serves as the input to the TVA layer, which focuses on both time and variable dimensions. Finally, the fully connected layer predicts the AUV's drift trajectory.

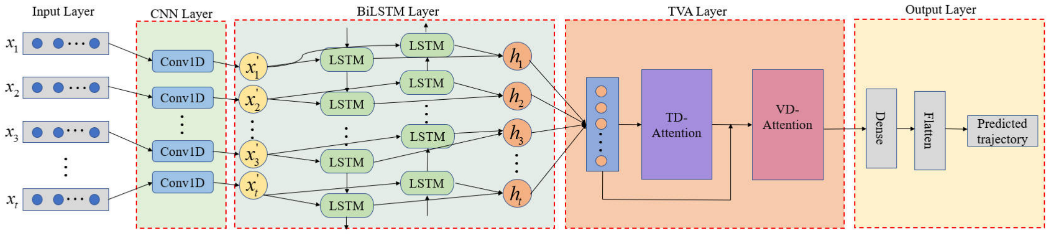


Figure 6. The structure of the BiLSTM-TVA model

3.6. NKOA-BiLSTM-TVA

This study uses the NKOA to optimize the parameters of the BiLSTM-TVA model. First, the upper and lower bounds of the hyperparameters of the BiLSTM-TVA model are determined, and then the NKOA optimizes the hyperparameters of the BiLSTM-TVA model. The optimized hyperparameters are used for model training and testing. The mean squared error (MSE) between the predicted trajectory and the real trajectory is calculated and fed back to the NKOA as the fitness value for algorithm iteration. Eventually, a set of optimal hyperparameters for the model is obtained. Figure 7 shows the detailed steps of the NKOA for optimizing the hyperparameters of the BiLSTM-TVA model.

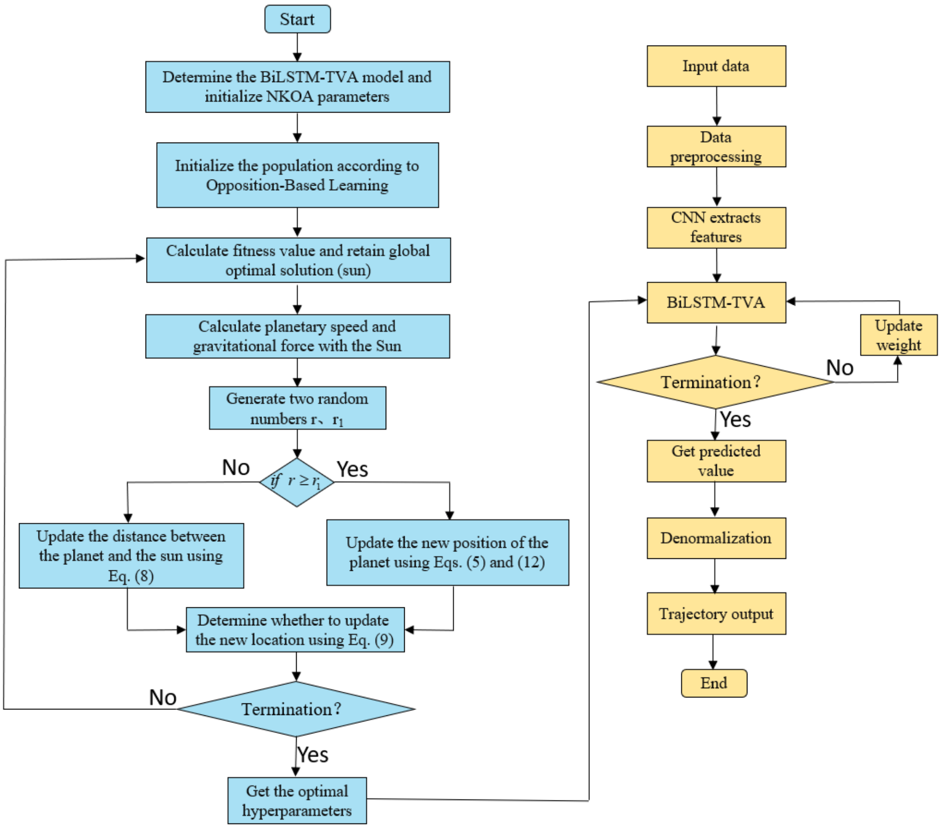

Start
Determine the BiLSTM-TVA model and initialize NKOA parameters
Initialize the population according to Opposition-Based Learning
Calculate fitness value and retain global optimal solution (sun)
Calculate planetary speed and gravitational force with the Sun
Generate two random numbers r、r1
No
if $r \geq r_1$
Yes
Update the distance between the planet and the sun using Eq. (8)
Update the new position of the planet using Eqs. (5) and (12)
Determine whether to update the new location using Eq. (9)
No
Termination?
Yes
Get the optimal hyperparameters
Input data
Data preprocessing
CNN extracts features
BiLSTM-TVA
Update weight
Termination?
No
Yes
Get predicted value
Denormalization
Trajectory output
End

Figure 7. NKOA process

The process of the NKOA is as follows:

(1) Determine and construct the BiLSTM-TVA model. Use the CNN model to capture the spatial feature dependency in the trajectory.

(2) Initialize the NKOA and its parameters, including the maximum number of iterations, the number of parameters to be optimized, the upper and lower bounds of the parameters, and the fitness function. Randomly initialize the algorithm population using opposition-based learning.

(3) Calculate the fitness values and retain the current global optimal fitness values and the current optimal parameters.

(4) Calculate the speed of the planets and the gravitational force between the planets and the Sun. Generate two random numbers r and r_1. If $r \geq r_1$, update the planet's position using Equations (5) and (12). Otherwise, update the distance between the planets and the Sun using Equation (8).

(5) Use Equation (9) to decide whether to update the planet's position or the distance between the planets and the Sun.

(6) Repeat steps (3)–(5) and check whether the algorithm has met the termination condition. The termination condition can be either reaching the number of iterations or the fitness value converging to a sufficiently small value. If the algorithm meets the termination condition, stop the optimization; otherwise, continue iterating the algorithm.

(7) After the algorithm ends, use the optimized hyperparameters to train the model on the training set. Then, validate the trained model on the test set to evaluate its performance.

(8) Save the model's predicted results, analyze them, and compare them with the prediction results of other models.

4. Experiments and Results

The section mainly analyzes the experimental results. First, it introduces the trajectory data preprocessing process and experimental evaluation metrics. Then, it analyzes the BiLSTM-TVA model parameter optimization results. Finally, we compare the experimental results of the proposed model with those of other models and analyze the results.

4.1. Data Preprocessing and Experimental Configuration

The AUV trajectory data used in the experiment come from the Autonomous Underwater Vehicle Monterey Bay Time-Series Dataset [38]. This dataset contains trajectory data measured by various sensors and positioning systems on the AUV. The trajectory data used in the experiments include the longitude, latitude, depth, pitch angle, and roll angle. The selected ocean environment data include the seawater salinity, ocean current speed, and oxygen concentration. First, the required trajectory features are separated from the original trajectory data, and outliers are removed from the trajectory data. Following this, the moving average method is employed to fill in missing values by taking the average of the data points within a certain time window. Lastly, the AUV trajectory is combined with marine environmental factors to construct the experimental dataset. To accelerate model convergence and improve prediction accuracy, normalization is applied to the data, scaling it within the [0, 1] range. The formula is as follows:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{27}$$

The dataset used consists of 10,000 records, with each record collected at one-minute intervals. The first 80% of the AUV trajectory data is used to train, while the remaining 20% is used to test. First, the model uses the NKOA to determine the optimal hyperparameters, and then it trains on the training set using these parameters. The sliding window size used in the experiment is 8. Since the longitude, latitude, and depth of the AUV can determine its position in three-dimensional space and the pitch angle and roll angle describe its attitude, the experiment in this paper predicts only these five trajectory variables. On the training set, the model's input and output sizes are (7991, 8, 8) and (7991, 5), respectively, and on the test set, they are (1991, 8, 8) and (1991, 5), respectively.

4.2. Model Evaluation Index

In order to objectively evaluate the prediction performance of the models, this paper uses two evaluation metrics in the experiment: root mean square error (RMSE) and mean absolute error (MAE). Their calculation formulas are as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)}^2 \tag{28}$$

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|\hat{y}_n - y_n| \tag{29}$$

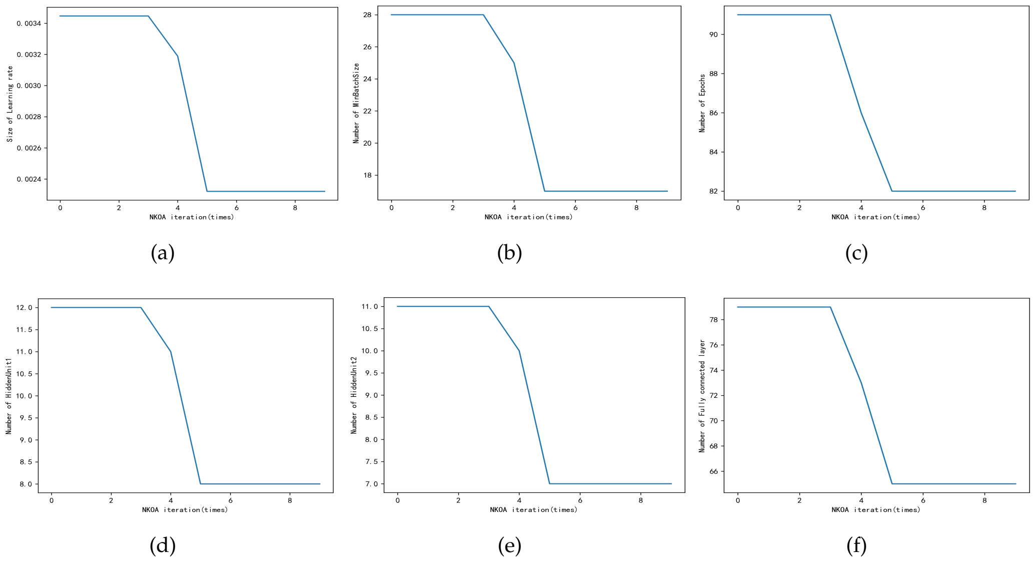
where N represents the total sample size of the AUV trajectory dataset, y_n represents the nth true trajectory value, and $\hat{y}_n$ is the trajectory value predicted by the model. For these two evaluation metrics, the closer their values are to 0, the smaller the prediction error, indicating greater accuracy of models.

4.3. Analysis of NKOA Hyperparameter Optimization Results

In this experiment, we used the NKOA to optimize the hyperparameters of the model. These hyperparameters included the learning rate, batch size, number of epochs, number of neurons in the first and second layers of the BiLSTM hidden layer, and number of fully connected layers. The number of iterations of the NKOA was set to 10. Figure 8 shows the optimization curve of the BiLSTM-TVA model parameters by the NKOA. From these optimization curves, we can observe the change in each parameter during each iteration. The horizontal coordinate represents the number of iterations of the NKOA, and the vertical coordinate represents the changes in model parameter values. Table 1 shows the range and final values of each parameter in the BiLSTM-TVA model. The parameters were optimized based on the fitness value of the NKOA. When the fitness value approaches its minimum and stability, it shows that the model parameters have been optimized to their best values. The fitness value of the NKOA stabilized after six iterations and finally settled at 0.00088. The learning rate converged from the range [0.001, 0.01] to 0.0023; the batch size from [8, 128] to 16; and the number of epochs from [10, 200] to 82. The number of neurons in the first hidden layer in the BiLSTM model converged from [1, 30] to 8; the number of neurons in the second hidden layer from [1, 30] to 7; and the number of neurons in the fully connected layer from [1, 100] to 65.

Table 1. Results of the NKOA's optimization of model parameters.

Hyperparameter	Range of Parameter	Optimal Value
Learning rate	[0.001, 0.01]	0.0023
Batch size	[8, 128]	16
Epoch	[10, 200]	82
Hidden layer 1	[1, 30]	8
Hidden layer 2	[1, 30]	7
Fully connected layer	[1, 100]	65



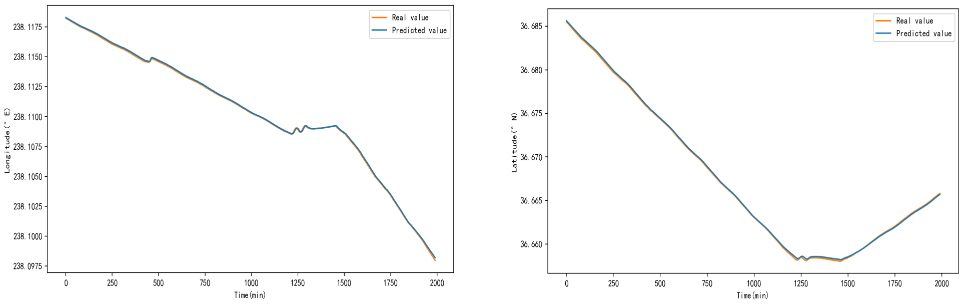
(a) (b) (c)

(d) (e) (f)

Figure 8. NKOA-BiLSTM-TVA optimal parameters: (**a**) optimization result for the learning rate; (**b**) optimization result for the batch size; (**c**) optimization result for the epochs; (**d**) number of nodes in the first hidden layer; (**e**) number of nodes in the second hidden layer; (**f**) number of fully connected layers.

4.4. Analysis of AUV Trajectory Prediction Results

After obtaining the optimized hyperparameters, the model first utilized the optimal hyperparameters to train on the training dataset and saved the trained model. Then, it used the trained model to perform model trajectory prediction testing on the test dataset. The experimental results for the NKOA-BiLSTM-TVA model on the test set were compared with those of the BiLSTM, DCNN-LSTM, CNN-BiLSTM-Attention, and TPA-BiLSTM models. Our experimental trajectory data were measured in a real ocean environment, so the model's predicted AUV trajectories could be compared with real data. For each model's prediction results, we demonstrated the curve fitting between the true and predicted values of the five trajectory variables and calculated the MAE and RMSE between them to verify the effectiveness of the predictions numerically. Since the experimental data were real and have been validated in practical applications, the reliability of the model's predictions could be confirmed. Figures 9–14 show the curve fitting between the true and the predicted values for the five trajectory variables, reflecting the model's prediction performance. Figure 9 shows the predicted longitude and latitude of the AUV's trajectory by the NKOA-BiLSTM-TVA model on the test set. Figures 10 and 11, respectively, compare the prediction results by several models for longitude and latitude. It can be observed in Figures 10 and 11 that all models correctly predicted the trend of longitude and latitude changes in the AUV's trajectory. However, some models without attention mechanisms did not predict a smooth curve. This is because these models do not use the feature dependency between trajectory points in their predictions. When the changes in longitude and latitude of the AUV's trajectory were small (as shown in the enlarged area in Figures 10 and 11), our model better fit the real trajectory due to its use of the feature dependency between trajectory variables. In contrast, models that do not consider the dependency between trajectory variables exhibited poorer prediction results.



(a) Longitude (b) Latitude

Figure 9. Trajectory predicted by the NKOA-BiLSTM-TVA model for longitude and latitude.

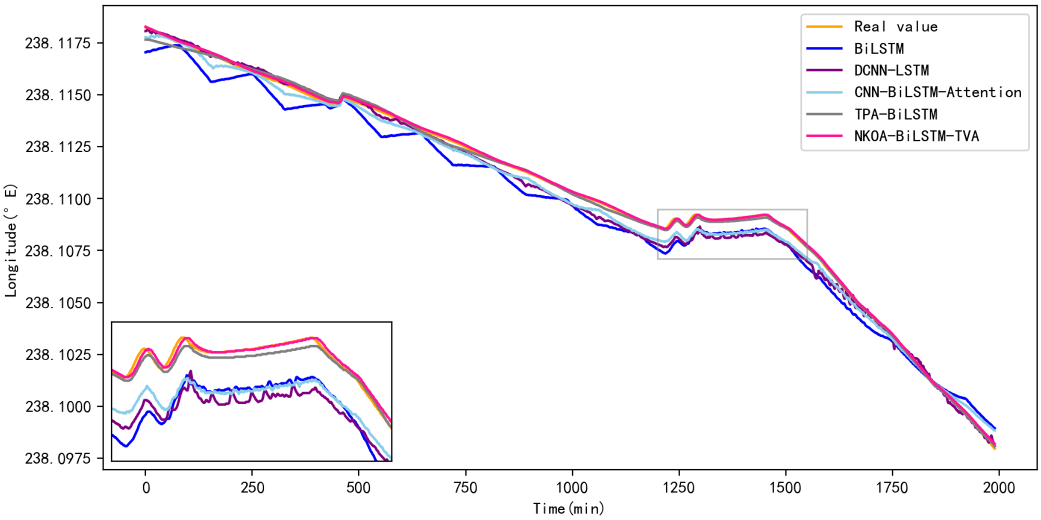


Figure 10. Trajectory predicted by the five models for longitude.

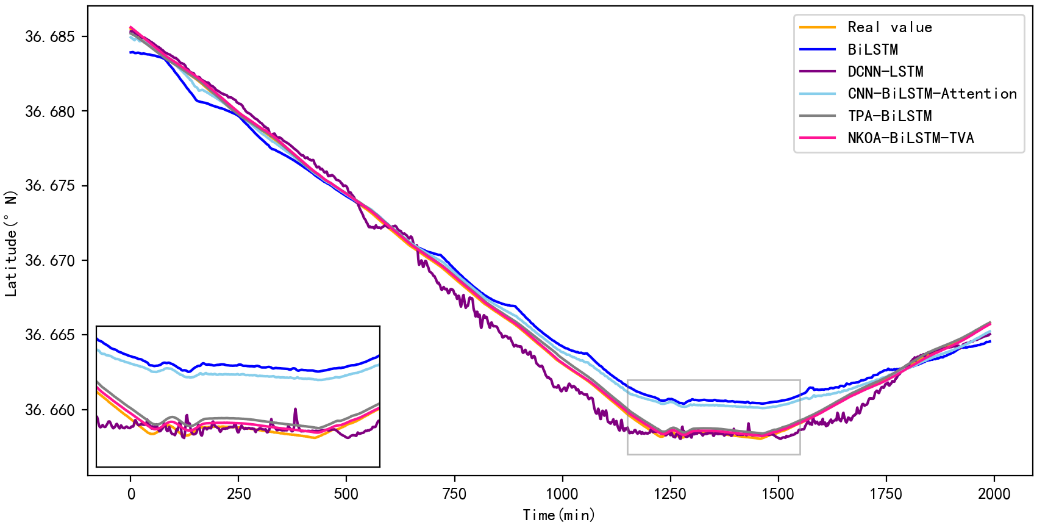
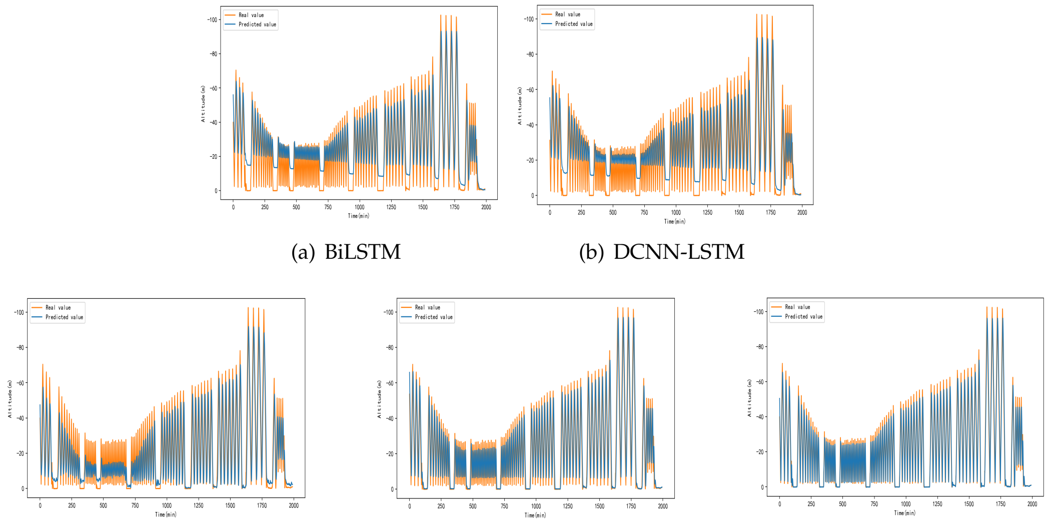


Figure 11. Trajectory predicted by the five models for latitude.

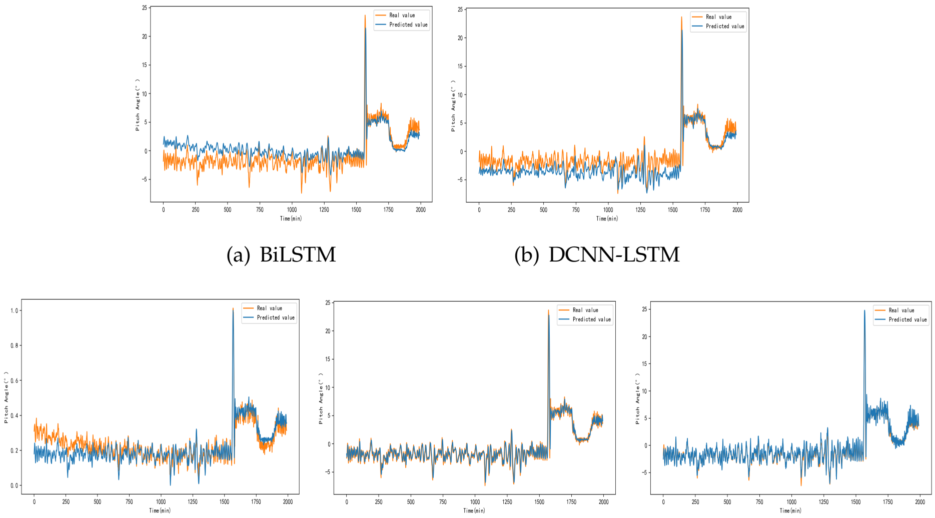
Figure 12 shows the prediction results for the AUV's altitude by the five models; Figure 13 shows the prediction results for the AUV's pitch angle; and Figure 14 shows the prediction results for the AUV's roll angle. Figure 15 shows a 3D image of the AUV's predicted drift trajectory by the NKOA-BiLSTM-TVA model on the test set.



(a) BiLSTM (b) DCNN-LSTM

(c) CNN-BiLSTM-Attention (d) TPA-BiLSTM (e) NKOA-BiLSTM-TVA

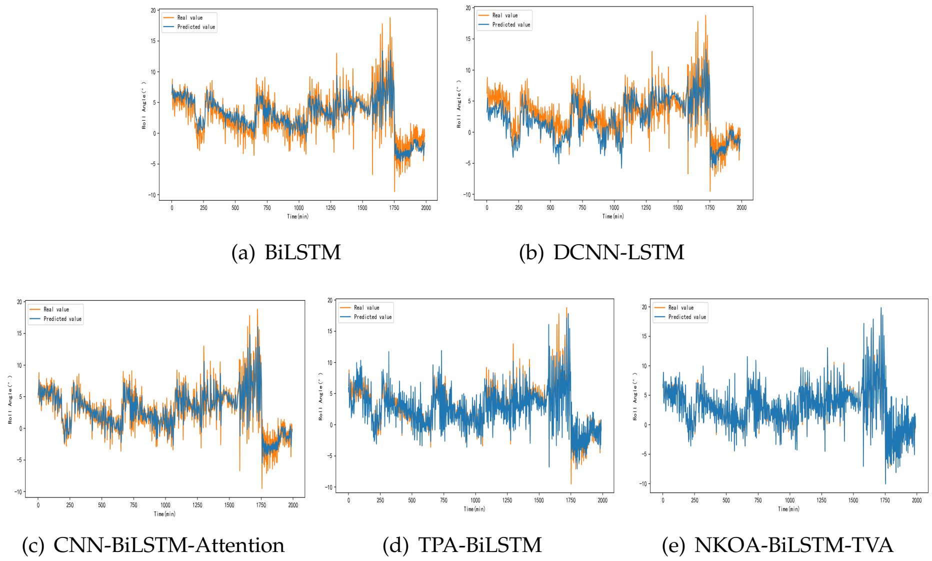
Figure 12. The predicted trajectories of the five models for the AUV's altitude: (**a**) Trajectory predicted by the BiLSTM model for the altitude; (**b**) Trajectory predicted by the DCNN-LSTM model for the altitude; (**c**) Trajectory predicted by the CNN-BiLSTM-Attention model for the altitude; (**d**) Trajectory predicted by the TPA-BiLSTM model for the altitude; (**e**) Trajectory predicted by the NKOA-BiLSTM-TVA model for the altitude.



(a) BiLSTM (b) DCNN-LSTM

(c) CNN-BiLSTM-Attention (d) TPA-BiLSTM (e) NKOA-BiLSTM-TVA

Figure 13. The predicted trajectories of the five models for the AUV's pitch angle: (**a**) Trajectory predicted by the BiLSTM model for the pitch angle; (**b**) Trajectory predicted by the DCNN-LSTM model for the pitch angle; (**c**) Trajectory predicted by the CNN-BiLSTM-Attention model for the pitch angle; (**d**) Trajectory predicted by the TPA-BiLSTM model for the pitch angle; (**e**) Trajectory predicted by the NKOA-BiLSTM-TVA model for the pitch angle.



(a) BiLSTM

(b) DCNN-LSTM

(c) CNN-BiLSTM-Attention

(d) TPA-BiLSTM

(e) NKOA-BiLSTM-TVA

Figure 14. The predicted trajectories of the five models for the AUV's roll angle: (**a**) Trajectory predicted by the BiLSTM model for the roll angle; (**b**) Trajectory predicted by the DCNN-LSTM model for the roll angle; (**c**) Trajectory predicted by the CNN-BiLSTM-Attention model for the roll angle; (**d**) Trajectory predicted by the TPA-BiLSTM model for the roll angle; (**e**) Trajectory predicted by the NKOA-BiLSTM-TVA model for the roll angle.

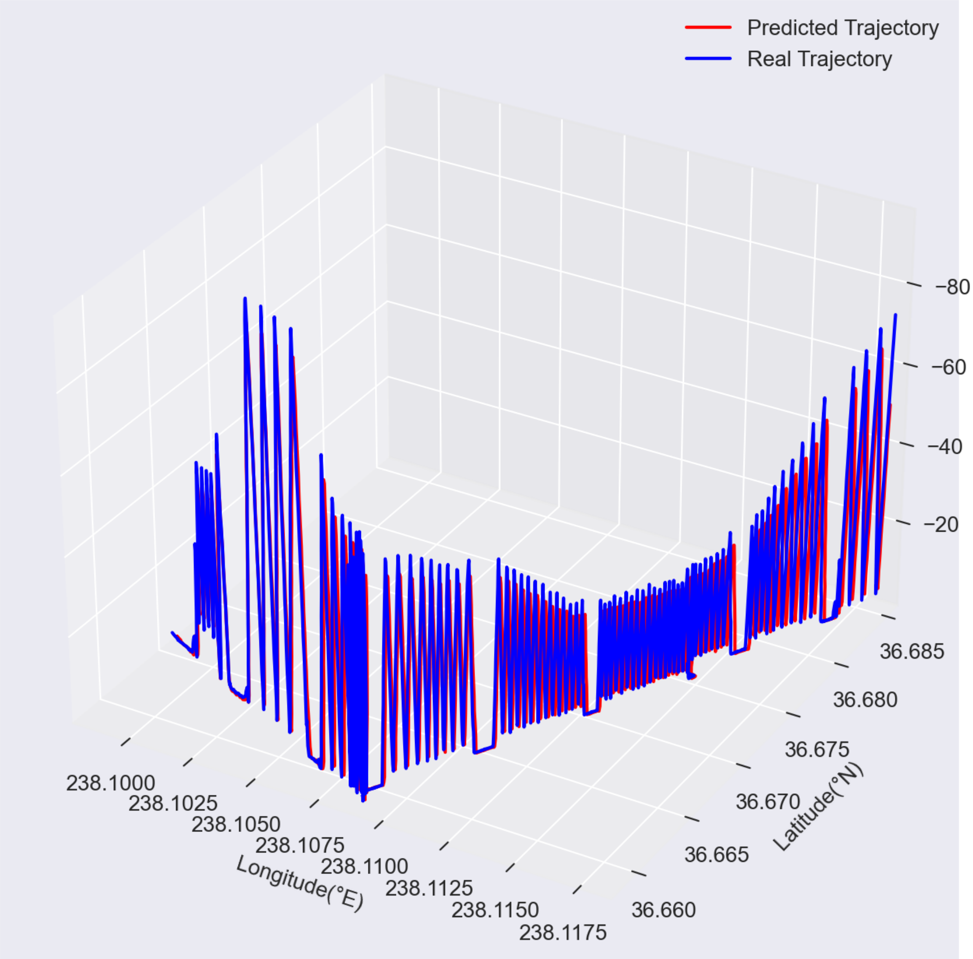


Figure 15. The NKOA-BiLSTM-TVA model's prediction of the AUV's 3D trajectory on the test set.

In Table 2, we present the calculated RMSE and MAE values between the predicted trajectory values and the true trajectory values on the AUV test set using Equations (28) and (29). For the four baseline models, the minimum MAE and RMSE for longitude are 0.0057 and 0.0072, respectively, while those of our model are 0.0038 and 0.0042. Similarly, for the four baseline models, the minimum MAE and RMSE for latitude are 0.0058 and 0.0069, while those of our model are 0.0044 and 0.0047. For the four baseline models, the minimum MAE and RMSE for altitude are 0.0816 and 0.0969, while those of our model are 0.0591 and 0.0703. For the four baseline models, the minimum MAE and RMSE for the pitch angle are 0.0218 and 0.0353, while those of our model are 0.0155 and 0.0205. Finally, for the four baseline models, the minimum MAE and RMSE for the roll angle are 0.0414 and 0.0543, while those of our model are 0.0212 and 0.0331.

Table 2. Comparison of RMSE and MAE values between all models on the AUV trajectory dataset.

Model	Metric	Longitude (°E)	Latitude (°N)	Altitude (m)	Pitch (°)	Roll (°)
BiLSTM	MAE RMSE	0.0332 0.0364	0.0364 0.0455	0.1314 0.1539	0.0539 0.0661	0.0551 0.0764
DCNN-LSTM	MAE RMSE	0.0193 0.0220	0.0224 0.0279	0.1023 0.1174	0.0524 0.0643	0.0711 0.0940
CNN-BiLSTM-Attention	MAE RMSE	0.0187 0.0240	0.0251 0.0355	0.0864 0.1083	0.0386 0.0513	0.0588 0.0835
TPA-BiLSTM	MAE RMSE	0.0057 0.0072	0.0058 0.0069	0.0816 0.0969	0.0218 0.0353	0.0414 0.0543
NKOA-BiLSTM-TVA	MAE RMSE	0.0038 0.0042	0.0044 0.0047	0.0591 0.0703	0.0155 0.0205	0.0212 0.0331

In Table 2, we can see that the order of these models in terms of prediction performance is NKOA-BiLSTM-TVA > TPA-BiLSTM > CNN-BiLSTM-Attention > DCNN-LSTM > BiLSTM. Moreover, Table 2 shows that the NKOA-BiLSTM-TVA model performs the best in predicting the five trajectory variables. Because the changes in the AUV drift trajectory's longitude and latitude are small, the TPA-BiLSTM model and NKOA-BiLSTM-TVA model have similar prediction performance, but NKOA-BiLSTM-TVA performs better in the other three variables. The results for the five models show that the model without an attention mechanism has significant trajectory jitter and a poorer fit to the actual AUV trajectory. The experimental results for the BiLSTM and DCNN-LSTM models compared to those for the other three models confirm the importance of attention mechanisms in predicting AUV drift trajectories. Models without attention mechanisms fail to give more weight to time steps that are critical for predicting future trajectories. The experimental results for the CNN-BiLSTM-Attention, TPA-BiLSTM, and NKOA-BiLSTM-TVA models confirm the impact of different attention mechanisms on AUV trajectory prediction. This validates the effectiveness of the TVA attention layer developed in this paper. It also confirms the importance of the dependency between different variables in predicting AUV trajectories.

4.5. Prediction Results for Ship Trajectories

To verify the effectiveness of the TVA layer, we continued to conduct experiments on the ship trajectory dataset. We used GRU, LSTM, and BiLSTM models to predict ship trajectories, and then added the TVA layer to these base models to form the following combined models: GRU-TVA, LSTM-TVA, and BiLSTM-TVA. In Table 3, we present the calculated RMSE and MAE values between the predicted trajectory values and the true trajectory values on the ship test set using Equations (28) and (29). The experimental results (Table 3) show that the combined models with the TVA layer perform better in terms of prediction accuracy compared to the base models. Additionally, among the three models, the BiLSTM model performs better than the GRU and LSTM models.

Table 3. Comparison of RMSE and MAE values on the ship trajectory dataset.

Model	**Longitude (°E)**		**Latitude (°N)**	
	MAE	**RMSE**	**MAE**	**RMSE**
GRU	0.0433	0.0668	0.0458	0.0721
GRU-TVA	0.0386	0.0528	0.0413	0.0653
LSTM	0.0302	0.0494	0.0393	0.0641
LSTM-TVA	0.0147	0.0392	0.0148	0.0417
BiLSTM	0.0209	0.0394	0.0183	0.0538
BiLSTM-TVA	0.0091	0.0159	0.0096	0.0166

5. Conclusions

To improve the performance of AUV trajectory prediction, this paper combined a CNN, BiLSTM, and a TVA layer to propose a high-accuracy AUV trajectory prediction method. This study used an improved Kepler optimization algorithm to optimize the model's hyperparameters. Our model used ocean environmental factors in the AUV trajectory prediction process, achieving multivariable AUV trajectory prediction. The improved NKOA shows an excellent parameter optimization effect. In the sixth iteration, the NKOA's fitness value reached a minimum of 0.00088, proving that the nonlinear factor accelerated the algorithm's convergence speed and enhanced the model's local search capability. We compared the prediction results of our model with those of the BiLSTM, DCNN-LSTM, CNN-BiLSTM-Attention, and TPA-BiLSTM models. The RMSE values of our model for the longitude, latitude, depth, pitch angle, and roll angle were 0.0042, 0.0047, 0.0703, 0.0205, and 0.0331, respectively. The prediction errors of our model were lower than those of other models, which proves the performance advantage of our model. In contrast, models without attention mechanisms have a very large jitter in their predicted trajectory curves, which also proves the importance of attention mechanisms in trajectory prediction. The TVA in our model effectively captured the dependency in both the time and variable dimensions of the trajectory, thereby improving the prediction accuracy by using the dependency between trajectory variables.

The limitation of this method lies in the need for a relatively large number of trajectory variables to extract the feature dependency. In future studies, we plan to combine the TVA layer with graph neural network models to extract richer trajectory features and achieve multi-step AUV trajectory prediction tasks while using fewer trajectory features to obtain better prediction results.

Author Contributions: Conceptualization, J.Y. (Jieen Yao) and J.Z.; methodology, J.Y. (Jieen Yao); software, J.Y. (Jieen Yao) and T.Z.; validation, J.Y. (Jieen Yao), J.Z., and T.Z.; formal analysis, J.Y. (Jieen Yao), J.Y. (Junzheng Yang), and C.Z.; investigation, J.Y. (Jieen Yao) and C.Z.; resources, J.Y. (Jieen Yao) and J.Y. (Junzheng Yang); data curation, J.Y. (Jieen Yao); writing—original draft preparation, J.Y. (Jieen Yao); writing—review and editing, J.Y. (Jieen Yao), J.Z., and T.Z.; visualization, J.Y. (Jieen Yao) and J.Z.; supervision, J.Z., J.Y. (Junzheng Yang), and C.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 (1) 2022–2025, the National Natural Science Foundation of China under Grant No. 52171310; and (2) 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: Since the data are part of an ongoing study, the dataset provided in this paper is not readily accessible. Please contact the corresponding author if you require the data provided in this article.

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

References

1. Zhou, J.; Si, Y.; Chen, Y. A review of subsea AUV technology. *J. Mar. Sci. Eng.* **2023**, *11*, 1119. [CrossRef]
2. Zhao, Y.; Hu, Z.; Du, W.; Geng, L.; Yang, Y. Research on Modeling Method of Autonomous Underwater Vehicle Based on a Physics-Informed Neural Network. *J. Mar. Sci. Eng.* **2024**, *12*, 801. [CrossRef]
3. Wang, L.; Zhu, D.; Pang, W.; Zhang, Y. A survey of underwater search for multi-target using Multi-AUV: Task allocation, path planning, and formation control. *Ocean. Eng.* **2023**, *278*, 114393. [CrossRef]
4. Xu, F.; Zhang, L.; Zhong, J. Three-Dimensional Path Tracking of Over-Actuated AUVs Based on MPC and Variable Universe S-Plane Algorithms. *J. Mar. Sci. Eng.* **2024**, *12*, 418. [CrossRef]
5. Liu, J.; Zhang, J.; Billah, M.M.; Zhang, T. ABiLSTM Based Prediction Model for AUV Trajectory. *J. Mar. Sci. Eng.* **2023**, *11*, 1295. [CrossRef]
6. Li, H.; Jiao, H.; Yang, Z. Ship trajectory prediction based on machine learning and deep learning: A systematic review and methods analysis. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107062. [CrossRef]
7. Liu, T.; Zhao, J.; Huang, J.; Li, Z.; Xu, L.; Zhao, B. Research on model predictive control of autonomous underwater vehicle based on physics informed neural network modeling. *Ocean. Eng.* **2024**, *304*, 117844. [CrossRef]
8. Perera, L.P.; Soares, C.G.; et al. Ocean vessel trajectory estimation and prediction based on extended Kalman filter. In Proceedings of the Second International Conference on Adaptive and Self-Adaptive Systems and Applications, Lisbon, Portugal, 21–26 November 2010; pp. 14–20.
9. Luo, X.; Wang, J.; Li, J.; Lu, H.; Lai, Q.; Zhu, X. Research on Ship Trajectory Prediction Using Extended Kalman Filter and Least-Squares Support Vector Regression Based on AIS Data. In Proceedings of the International Conference on Intelligent Transportation Engineering, Indianapolis, IN, USA, 19–22 September 2021, pp. 1123–1131.
10. Song, S.; Liu, J.; Guo, J.; Wang, J.; Xie, Y.; Cui, J.H. Neural-network-based AUV navigation for fast-changing environments. *IEEE Internet Things J.* **2020**, *7*, 9773–9783. [CrossRef]
11. Masini, R.P.; Medeiros, M.C.; Mendes, E.F. Machine learning advances for time series forecasting. *J. Econ. Surv.* **2023**, *37*, 76–111. [CrossRef]
12. 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.
13. Fernando, T.; Denman, S.; Sridharan, S.; Fookes, C. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural Netw.* **2018**, *108*, 466–478. [CrossRef]
14. Dai, S.; Li, L.; Li, Z. Modeling vehicle interactions via modified LSTM models for trajectory prediction. *Ieee Access* **2019**, *7*, 38287–38296. [CrossRef]
15. Lin, L.; Li, W.; Bi, H.; Qin, L. Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms. *IEEE Intell. Transp. Syst. Mag.* **2021**, *14*, 197–208. [CrossRef]
16. Peng, Y.; Zhang, G.; Shi, J.; Xu, B.; Zheng, L. SRAI-LSTM: A social relation attention-based interaction-aware LSTM for human trajectory prediction. *Neurocomputing* **2022**, *490*, 258–268. [CrossRef]
17. Li, J.; Li, W. Auv 3d trajectory prediction based on cnn-lstm. In Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, China, 7–10 August 2022; pp. 1227–1232.
18. Jameer, S.; Syed, H. A DCNN-LSTM based human activity recognition by mobile and wearable sensor networks. *Alex. Eng. J.* **2023**, *80*, 542–552. [CrossRef]
19. Zhu, H.; Liu, C.; Wu, H. A prediction method of seedling transplanting time with DCNN-LSTM based on the attention mechanism. *Agronomy* **2022**, *12*, 1504. [CrossRef]
20. Chen, S.; Chen, B.; Shu, P.; Wang, Z.; Chen, C. Real-time unmanned aerial vehicle flight path prediction using a bi-directional long short-term memory network with error compensation. *J. Comput. Des. Eng.* **2023**, *10*, 16–35. [CrossRef]
21. Jia, P.; Chen, H.; Zhang, L.; Han, D. Attention-LSTM based prediction model for aircraft 4-D trajectory. *Sci. Rep.* **2022**, *12*, 15533. [CrossRef]
22. Yang, L.; Li, S.; Zhu, C.; Zhang, A.; Liao, Z. Spatio-temporal correlation-based multiple regression for anomaly detection and recovery of unmanned aerial vehicle flight data. *Adv. Eng. Informatics* **2024**, *60*, 102440. [CrossRef]
23. Sun, J.; Zeng, H.; Ye, K. Short-Term Exhaust Gas Temperature Trend Prediction of a Marine Diesel Engine Based on an Improved Slime Mold Algorithm-Optimized Bidirectional Long Short-Term Memory—Temporal Pattern Attention Ensemble Model. *J. Mar. Sci. Eng.* **2024**, *12*, 541. [CrossRef]
24. Abdel-Basset, M.; Mohamed, R.; Azeem, S.A.A.; Jameel, M.; Abouhawwash, M. Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowl.-Based Syst.* **2023**, *268*, 110454. [CrossRef]
25. Abid, M.; Belazzoug, M.; Mouassa, S.; Chanane, A.; Jurado, F. Optimal power flow of thermal-wind-solar power system using enhanced Kepler optimization algorithm: Case study of a large-scale practical power system. *Wind. Eng.* **2024** , 0309524X241229206. [CrossRef]
26. Abdel-Basset, M.; Mohamed, R.; Alrashdi, I.; Sallam, K.M.; Hameed, I.A. CNN-IKOA: convolutional neural network with improved Kepler optimization algorithm for image segmentation: experimental validation and numerical exploration. *J. Big Data* **2024**, *11*, 13. [CrossRef]
27. Mohamed, R.; Abdel-Basset, M.; Sallam, K.M.; Hezam, I.M.; Alshamrani, A.M.; Hameed, I.A. Novel hybrid kepler optimization algorithm for parameter estimation of photovoltaic modules. *Sci. Rep.* **2024**, *14*, 3453. [CrossRef]

28. Tizhoosh, H.R. Opposition-based learning: a new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria , 28–30 November 2005; Volume 1, pp. 695–701.
29. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
30. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3285–3292.
31. Karim, F.; Majumdar, S.; Darabi, H.; Harford, S. Multivariate LSTM-FCNs for time series classification. *Neural Netw.* **2019**, *116*, 237–245. [CrossRef] [PubMed]
32. Sagheer, A.; Kotb, M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* **2019**, *323*, 203–213. [CrossRef]
33. 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*.
34. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 11106–11115.
35. Du, D.; Su, B.; Wei, Z. Preformer: predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
36. Mondoloni, S.; Rozen, N. Aircraft trajectory prediction and synchronization for air traffic management applications. *Prog. Aerosp. Sci.* **2020**, *119*, 100640. [CrossRef]
37. Zhang, K.; Li, L. Explainable multimodal trajectory prediction using attention models. *Transp. Res. Part Emerg. Technol.* **2022**, *143*, 103829. [CrossRef]
38. Scholin, C. *Autonomous Underwater Vehicle Monterey Bay Time Series—AUV Dorado from AUV Dorado in Monterey Bay from 2003-2099 (C-MORE Project, Prochlorococcus Project)*; Biological and Chemical Oceanography Data Management Office (BCO-DMO) : Woods Hole, MA, USA, 2011; Version Date 2011-01-26 [if applicable, indicate subset used].

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.