Genetic Algorithm and LSTM Based Approach for Diver Search and Path Prediction

Xiangyang Liu, Zhengyang Wang, Jintong Xiong and Zhaoyu Liu *

College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University, Nanjing, China

* Corresponding Author Email: 1206212696@qq.com

Abstract. With the rapid development of submersible technology, exploring the deep sea is no longer a distant dream. In this paper, a set of safety procedures for manned submersibles is designed to ensure that the main ship can respond in time and launch search and rescue in case of loss of connection. The digital elevation model and Kalman filter algorithm are used to process the deep-sea data, combined with the Monte Carlo method to estimate the position of the submersible, and the water depth of 2000m is taken as the working depth. The genetic algorithm is used to optimize the SAR strategy, and environmental factors are considered to affect the sailing speed. For the case of multi-diver lost connection, CNN and LSTM models are combined to predict the 3D path to avoid the risk of collision. These methods will significantly improve the safety of the submersible and the efficiency of search and rescue and promote the sustainable development of deep-sea exploration.

Keywords: Kalman filtering algorithm; genetic algorithm; LSTM.

1. Introduction

With the rapid development of submersible technology and the continuous advancement of deep-sea exploration, mankind's desire to explore the mysterious world of the deep sea is increasing day by day [1]. However, the safety of manned submersibles has become the focus of attention, especially after the implosion of the Titan in 2023 [2]. To address this challenge, the aim of this paper is to design an efficient and safe submersible search and rescue (SAR) procedure to ensure that the primary vessel can respond quickly and conduct SAR operations in the event of a lost submersible. Through the use of digital elevation models, Kalman filter algorithms, Monte Carlo methods, and genetic algorithms, this paper explores how to improve the efficiency and accuracy of SAR in order to promote the safety of submersibles and the sustainable development of the field of deep-sea exploration.

2. Submersible Position Prediction

2.1. Kalman Filter Algorithm

Considering that the submersible sailing on the seabed is multidimensionally stressed, we establish a dynamics model based on the submersible, according to Newton's second law $\vec{F} = m \cdot \vec{a}$, where F is the net force acting on the submersible and the acceleration is a function of time, and because $a(t) = \frac{d^2x}{dt^2}$,we can therefore establish the equations of motion:

$$m\frac{d^2\vec{r}}{dt^2} = \vec{F}_{prop} + \vec{F}_{drag} + \vec{F}_{buoyancy} + \vec{F}_{gravity} \tag{1}$$

Where r is the position vector of the submersible. The equation for the resistance of the submersible in a given direction is given below:

$$F_{drag} = -\frac{1}{2}\rho(v+u)^2 C_D M\hat{v} \tag{2}$$

Divide the force into motion in the x (horizontal), y (vertical), and z (vertical) directions. For each of the x, y, and z directions, it can be decomposed separately:

$$m \cdot \frac{d^2x}{dt^2} = F_{prop,x} - F_{drag} + F_{error} \tag{3}$$

$$m \cdot \frac{d^2y}{dt^2} = F_{prop,y} - F_{drag} + F_{error} \tag{4}$$

$$m \cdot \frac{d^2z}{dt^2} = F_{prop,z} + F_{buoyancy} - F_{gravity} - F_{drag} \tag{5}$$

For the kinetic equations described above, we solved them using the fourth order Lunger-Kutta method to obtain the predicted trajectories of the submersible in three-dimensional space over a period of time.

By observing the model results, we found that the trajectories composed of points derived from the kinetic model were haphazard, so we began to reflect that the kinetic model is from a physical point of view, simulating continuous changes in physical processes, while the underwater travel process of the submersible must be a travel trajectory composed of discrete points. Given this, we came up with the Kalman filter. The Kalman filter is a recursive algorithm for estimating the state of a system that uses measurements of the system and the dynamic equations of the model to estimate the implied state of the system and provides an optimal estimate of the state through recursive updating [3]. In the following we use the Kalman filter to solve the discrete dynamic problem of the diver's position over time.

Here we only consider the linear motion of the submersible without considering the rotation factor, then the state vector can be expressed as:

$$\vec{x} = \left[x, y, z, v_x, v_y, v_z\right]^T \tag{6}$$

Where x, y, and z denote the position of the submarine, and below we need to build state equations to describe how the state of the submarine evolves as it changes from one moment to the next:

$$\vec{x}_k = \vec{A} \cdot \vec{x}_{k-1} + \vec{B} \cdot \vec{v}_k + \vec{w}_k \tag{7}$$

Where w_k is the process noise and satisfies the normal distribution $N(0, Q)$.

The current state x_k of the system is then transformed into the measured value z_k in the measurement space:

$$\vec{z}_k = \vec{H} \cdot \vec{x}_k + \vec{v}_k \tag{8}$$

Where v_k is the measurement noise that satisfies the normal distribution $N(0, R)$.

Kalman filtering expresses the uncertainty of the state estimates through the state covariance matrix P_k. A larger P_k indicates a larger uncertainty. By taking noise and uncertainty into account, we are able to estimate the position and trajectory of the submersible more accurately, thus providing strong support for marine research and resource exploration.

For the computational matrix A, it is used to describe how the state is transferred from one moment to the next, which usually includes the update of states such as position, velocity and attitude. For a simplified linear system, A can be expressed as:

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Where Δt denotes the time step, this matrix is used to describe the linear update of position, velocity and attitude.

For the control input matrix B and the control vector v_k, this matrix describes how the control input vector u is associated with state changes in the system dynamics model. Let us consider a simplified example to simulate the computational matrix B. Assuming that we only consider the linear motion of the submersible, we can simplify the control vector to the control of the submersible's linear motion in three axes:

$$\vec{v}_k = \left[F_{prop,x}, F_{prop,y}, F_{prop,z}\right]^T \tag{10}$$

Matrix B has dimension bit 6×3 and represents the relationship between the state variables and the control variables. In this case, C is calculated as:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

In order to measure the matrix H is used to map the state variables to the measurement space, i.e., the state variables are converted to the expectations of the sensor measurements. The choice of H depends on the measurement system and the sensor of the submersible. Typically, H is a matrix whose elements represent the relationship between state variables and measured values. For simplicity, assuming that there is a gauge that directly measures position coordinates (x, y, z), then H will be a unit matrix.

2.2. Location Prediction Results

Using the Kalman filter to filter out process noise and observation noise, and through data visualization, we can see that the Kalman filter is able to effectively estimate the position of the submersible on the seafloor based on the observation data when subjected to sustained propulsive forces and sudden interruptions of propulsive forces.

As shown in Fig. 1, in the case of sustained propulsion, the Kalman filter is able to estimate the position of the submersible using the dynamic model of the system and the observation data. It will make state predictions based on observations and the system model and update the state based on new observations. This iterative process gradually reduces the uncertainty and provides more stable position estimation results for the main vessel.

As shown in Fig. 2, when the submersible is suddenly interrupted in propulsion, the Kalman filter is able to detect the effects of the propulsion interruption and adaptively adjust the state estimate to minimize the effect of the interruption on the position estimate. This capability allows the Kalman filter to provide the host vessel with a reliable position estimate of the submersible even in the face of sudden changes in the submersible system. This allows for more efficient search and rescue operations.

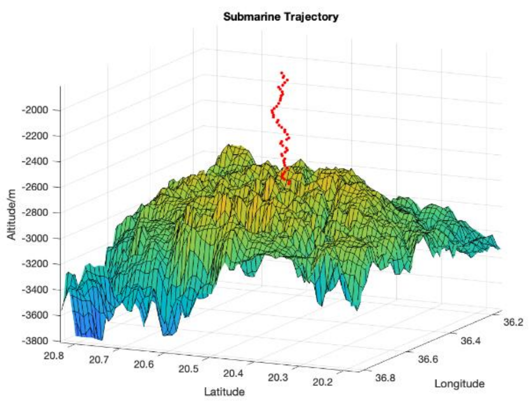

Fig. 1 Predicted trajectory of the submersible without loss of propulsion

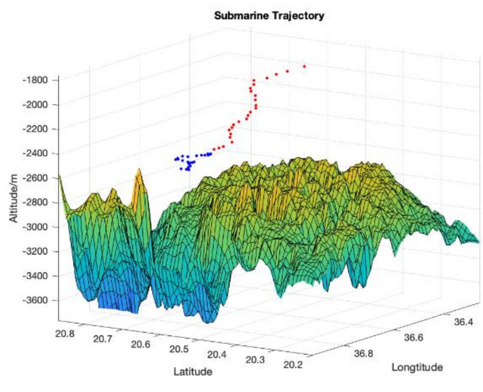

Fig. 2 Predicted trajectory of submersible loss of power

3. Search for Lost Submarine

To better analyze the problem, we use the analogy of the traveler problem. In this problem, the search space is defined as the area in which the submersible may exist, each "city" can be analogized as a possible location point of the submersible, and each individual represents a search path, i.e., the order in which the search points are visited, for a number of possible location points simulated by Monte Carlo within the search space [4]. The fitness of an individual is determined by the efficiency and coverage of the search path, which can be measured in terms of search path length, search time, or probabilistic model. The goal is to minimize the search time or maximize the probability of finding a submarine.

Assuming that the set of all possible location points is $C = \{c_1, c_2, \dots, c_n\}$, it is necessary to compute the distance between each of the possible location points to obtain a distance matrix D. Use the following formula to compute the Euclidean distance between two location points:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} \tag{12}$$

Where x_i, y_i, z_i and x_j, y_j, z_j are the coordinates of the two search points respectively. Each search point has environmental factors associated with it, such as the direction of ocean currents and the depth of the sea. Ocean currents can speed up or slow down the movement of the search vessel. The effect of ocean currents can be represented by a velocity vector $v_{current}(x_i, y_i, z_i)$. The effects of other factors including weather conditions $w(x_i, y_i, z_i)$and sea depth $d(x_i, y_i, z_i)$ may affect the search time.

Our goal is to find an alignment $P = (p_1, p_2, \dots, p_n)$ that visits all cities, where $p_i$is a location point in the set C and each location point is visited exactly once, in order to minimize the total travel time, which is given by the following equation:

$$T(P) = \sum_{i=1}^{n-1} \frac{d_{i,i+1}}{s + |\vec{v}_{current}(x_i, y_i, z_i)| \cdot \cos(\theta_{i,i+1})} \cdot w(x_i, y_i, z_i) \cdot d(x_i, y_i, z_i) \tag{13}$$

Where $d_{i,i+1}$ is the distance between search point i and $i+1$, s is the base speed of the search vessel or UAV, $\theta_{i,i+1}$ is the angle between the direction of movement from point i to $i+1$ and the direction of the current, $|v_{current}(x_i, y_i, z_i)|$ is the current velocity at point i the current velocity at point i, and $w(x_i, y_i, z_i)$ and $d(x_i, y_i, z_i)$ are the weather coefficients and depth coefficients at point i, respectively.

In the following, a genetic algorithm is applied to find a set of the most suitable alignments P [5]. Individuals are denoted as sequences of search points, representing search paths. By making 50 iterations of the possible trajectories after the loss of the submersible, Fig. 3 below shows the change of the length of the search path after 50 iterations, and it can be found that with the increase of the number of iterations, the model's planning of the search path is getting shorter and shorter, which greatly improves the efficiency of search and rescue.

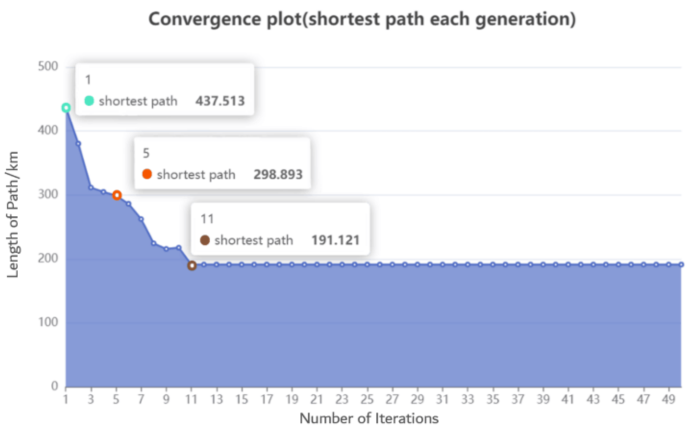

Fig. 3 Shortest path convergence graph

We select three of the iteration results to show, Fig. 4 below shows the results of the 1st, 5th, and 11th iterations from left to right, respectively.

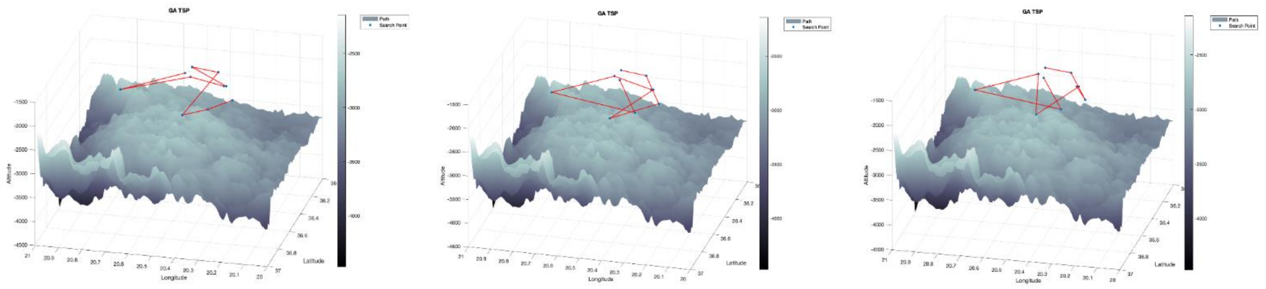

Fig. 4 Iteration diagram of the path of TSP genetic algorithm

As we know, it takes some time for the rescue ship to arrive after the submarine is lost, the main ship needs to predict the position of the submarine and search according to the path derived from the genetic algorithm. On the way to search, the active sonar system is turned on to release pulse signals to search and locate the submersible, and a certain number of sonar buoys are placed in the vicinity of each search point to maximize the search efficiency of the hydroacoustic system. The main ship maintains communication with the rescue ship to share hydroacoustic information. If the location of the submersible is found, it will be reported immediately, and the rescue ship will go directly to the wreck and use the ROV to dive to the submersible to fix it and then salvage it with the rescue ship's winch.

If the sonar system does not find the position information of the submersible in the predicted sea area, it will conduct a spiral search centered on the center of the predicted range of the lost point until it finds the position of the submersible. Factors considered in this search include the primary vessel, the rescue vessel, the ROV, the active sonar system, and the lost submersible.

Fig. 5 below shows, in order, three scenarios where the prediction error for the spiral search is in the unacceptable range, the best case within the acceptable range, and the worst case within the acceptable range.

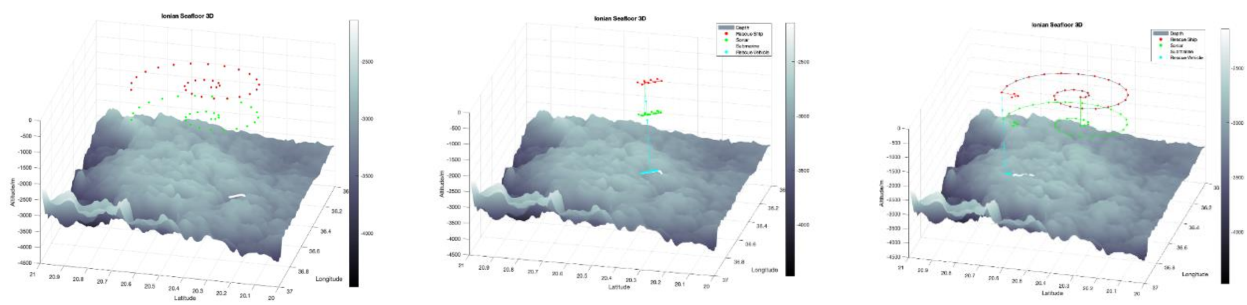

Fig. 5 Search and rescue schematic

4. Submersible Path Prediction

For searching and rescuing multiple submersibles in the same ocean, we believe that the problem lies in utilizing both spatial and temporal features. We utilize deep learning methods, specifically combining CNN and LSTM, to predict the 3D paths of submarines. Specifically, the deep learning model is able to take current seafloor images and past submarine locations as inputs and output future submarine path coordinates (typically including X, Y, and Z coordinates). We assume that the submarine's path is continuous in 3D space, and the inputs to the model include spatial features (e.g., seafloor images) and temporal features (e.g., past submarine positions).

We use the mean square error (MSE) as a loss function to measure the difference between the predicted path and the true path. The input data to the model is a sequence of seafloor images $I_{spatical}$, a sequence of past submarine positions $I_{temporal}$, and the output data is a sequence of coordinates of the path of future submarines Y. The data from multiple submarines can be passed to the model as a batch. Each batch contains data from multiple submarines in order to process multiple samples simultaneously.

The goal of the model is to minimize the mean square error (MSE) between the predicted path and the true path:

$$MSE\left(Y,\hat{Y}\right)=\frac{1}{N}\sum_{I=1}^{N}||Y_i-\hat{Y_\iota}||^{\,2} \tag{14}$$

Where $\hat{Y}$ is the path predicted by the model. Meanwhile, in order to avoid collisions, a collision penalty term is to be introduced into the loss function to encourage the model to generate paths without collisions, which can be achieved by modifying the loss function. Suppose $D\left(\widehat{Y_\iota},\widehat{Y_j}\right)$ is the collision detection function between the $i-th$ submarine and the $j-th$ submarine, and it returns a non-negative value indicating the degree of collision. You can then add a collision penalty term to the loss function:

$$L_{with\ Collsion\ Penalty}=MSE\left(Y,\hat{Y}\right)+\alpha\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}D\left(\widehat{Y_\iota},\widehat{Y_j}\right) \tag{15}$$

Where α is the weight of the collision penalty, which is used to control the degree of impact of the collision penalty. $D\left(\widehat{Y_\iota},\widehat{Y_j}\right)$ can be used to measure the distance between two paths by Euclidean distance.

Next, we use CNN to process the seabed image sequence to extract the spatial features of the image [6]. The output of the CNN will be a series of feature maps, each capturing a different spatial feature. This step can be represented as:

$$Spatical\ Features=CNN\left(I_{spatial}\right) \tag{16}$$

Where CNN denotes the operation of convolutional neural network including convolutional layer, pooling layer, and fully connected layer. The convolutional layer uses convolutional operations to extract local features of the input seabed image sequence. The mathematical formula for the convolution operation is as follows:

$$(f*g)(t)=\int f(\tau)g(t-\tau)d\tau \tag{17}$$

Where f is the input image, g is the convolution kernel (filter), τ is the integration variable, and $*$ denotes the convolution operation.

The pooling layer is used to reduce the dimensionality of the feature map and reduce the computational complexity. Maximum pooling is a commonly used pooling operation:

$$(Max\ Pooling)(x,y)=\max_{i,j}Pool\,[i,j] \tag{18}$$

The pooled feature representation is then used as part of the model for subsequent prediction. Next, we use an LSTM to process past diver position sequences to capture dynamic features in the time series. The output of the LSTM will be a series of hidden states, where each hidden state contains an encoding of a past position sequence. The mathematical modeling of this step can be expressed as:

$$Temporal\ Features=LSTM(I_{temporal}) \tag{19}$$

LSTM is a recurrent neural network (RNN) variant suitable for processing time-series data that captures long-term dependencies in sequential data [7]. LSTM consists of input gates, forgetting gates, output gates, and cell states. In this problem, input gates represent new search information, such as real-time data from sensors, including sonar signals, underwater images, and so on. The forgetting gate represents forgetting old search information in this problem, i.e., information about the target location or path is extracted from previous search states. The cell state is updated and passed on at each time step, and it can be viewed as an internal state of the search model for capturing long-term dependencies in the search process. The cell state is updated based on input gates and forgetting gates.

The output gate here represents the extraction of information about the target location or path from the search state. The mathematical formulation is as follows.

After obtaining the spatial and temporal features, we fuse them to generate a joint feature representation. The fusion can be done by splicing or weighted fusion and is modeled as follows:

$$Combined\ Features = Fusion(Spatial\ Features, Temporal\ Features) \tag{20}$$

Finally, we construct a predictive model that accepts the joint feature representation as input and outputs future diver path coordinates. The mathematical modeling is as follows:

$$Y_{predicted} = Predictor(Combined\ Features) \tag{21}$$

Where $Predictor$ denotes the operation of the predictive model.

The training phase of the model involves using the training data (including $I_{spatial}$, $I_{temporal}$, and Y) to train the predictive model. During training, we use the mean square error to measure the error between the predicted path and the true path. The backpropagation algorithm and optimizer are used to update the weights and biases of the model sand minimize the loss function. Once the model training is complete, we can use evaluation data to assess the performance of the model.

By combining CNN and LSTM, we are able to predict the 3D path of the submersible more accurately. This method fully utilizes both spatial and temporal features to improve the accuracy of path prediction.

After deep learning model training, the results are shown in Fig. 6, where each color represents a diver and the dots connecting various colors indicate a trajectory:

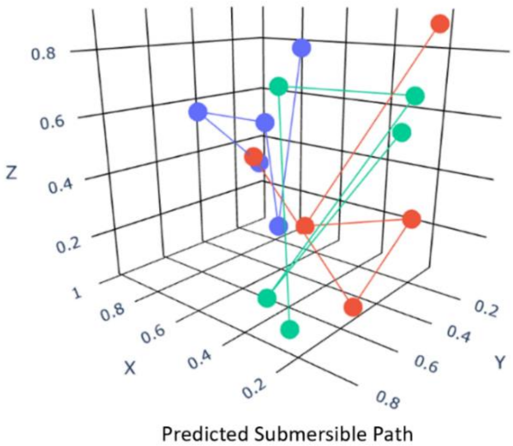

Fig. 6 Multiple Submersibles predicted path

5. Summary

The aim of this study is to design an efficient and safe submersible search and rescue (SAR) program to enhance the safety of submersibles and to promote the sustainable development of the deep-sea exploration field. Through the combined use of digital elevation model, Kalman filtering algorithm, Monte Carlo method and genetic algorithm, we have established a complete set of search and rescue mechanism, which effectively improves the efficiency and accuracy of search and rescue.

Our results show that the position prediction model using a combination of Kalman filtering algorithm and Monte Carlo method can effectively deal with the case of a lost submersible, while the genetic algorithm provides a choice of the optimal SAR path for the main vessel. Meanwhile, the combination of CNN and LSTM model predicts the paths of multiple submersibles, which further improves the SAR efficiency and accuracy.

With the results of this research, we provide new ideas and methods for the technological development and application in the field of SAR for submersibles, which is expected to contribute to the improvement of the safety level of submersibles and the sustainable development of the deep-sea exploration business. We hope that this study will play an active role in future practice of SAR on submersibles, safeguard the safety of deep-sea explorers, and promote the progress and development of deep-sea science.

References

[1] Stel J H. Exploring and exploiting deep ocean space[J]. Ocean Literacy: Understanding the Ocean, 2021: 65-92.

[2] Chen D, Qiao Y, Sun Y, et al. Human reliability assessment and risk prediction for deep submergence operating system of manned submersible under the influence of cognitive performance[J]. Ocean Engineering, 2022, 266: 112753.

[3] Khodarahmi M, Maihami V. A review on Kalman filter models[J]. Archives of Computational Methods in Engineering, 2023, 30(1): 727-747.

[4] Hemeida M G, Alkhalaf S, Senjyu T, et al. Optimal probabilistic location of DGs using Monte Carlo simulation based different bio-inspired algorithms[J]. Ain Shams Engineering Journal, 2021, 12(3): 2735-2762.

[5] Behera N, Jeevitesh M S, Jose J, et al. Higher accuracy protein multiple sequence alignments by genetic algorithm[J]. Procedia Computer Science, 2017, 108: 1135-1144.

[6] Anokye M, Cui X, Yang F, et al. CNN multibeam seabed sediment classification combined with a novel feature optimization method[J]. Mathematical Geosciences, 2024, 56(2): 279-302.

[7] Song X, Liu Y, Xue L, et al. Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model[J]. Journal of Petroleum Science and Engineering, 2020, 186: 106682.