

Article

Rapid Modeling of the Sound Speed Field in the South China Sea Based on a Comprehensive Optimal LM-BP Artificial Neural Network

Jin Huang ¹, Yu Luo ^{1,*}, Jian Shi ¹, Xin Ma ², Qian-Qian Li ^{1,3} and Yan-Yi Li ⁴

¹ College of Geodesy and Geomatics, Shandong University of Science and Technology, Qingdao 266500, China; huangjin@sdust.edu.cn (J.H.); shijian_sdust@163.com (J.S.); lqq_sdust@163.com (Q.-Q.L.)

² School of Information Science and Engineering, Shandong University, Jinan 250100, China; xinma_sdu@163.com

³ State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China

⁴ College of Surveying and Geo-Informatics, Tongji University, Shanghai 200092, China; LiYanyi19981104@163.com

* Correspondence: luoyu@sdust.edu.cn; Tel.: +86-138-0123-8460

Abstract: Ocean sound speed is an essential foundation for marine scientific research and marine engineering applications. In this article, a model based on a comprehensive optimal back propagation artificial neural network model is developed. The Levenberg–Marquardt algorithm is used to optimize the model, and the momentum term, normalization, and early termination method were used to predict the high precision marine sound speed profile. The sound speed profile was described by five indicators: date, time, latitude, longitude, and depth. The model used data from the CTD observation dataset of scientific investigation over the South China Sea (2009–2012) (108°–120° E, 6°–8° N), which includes comprehensive scientific investigation data from four voyages. The feasibility of modeling the sound speed field in the South China Sea is investigated. The proposed model uses the momentum term, normalization, and early termination in a traditional BP artificial neural network structure and mitigates issues with overtraining and difficulty when determining the BP neural network parameters. With the LM algorithm, a fast-modeling method for the sound field effectively achieves the precision requirement for sound speed prediction. Through the prediction and verification of the data from 2009 to 2012, the newly proposed optimized BP network model is shown to dramatically reduce the training time and improve precision compared to the traditional network model. Results showed that the root mean squared error decreased from 1.7903 m/s to 0.95732 m/s, and the training time decreased from 612.43 s to 4.231 s. Finally, the sound ray tracing simulations confirm that the model meets the accuracy requirements of acoustic sounding and verify the model’s feasibility for the real-time prediction of the vertical sound speed in saltwater bodies.

Keywords: sound speed profile; CTD data; Levenberg–Marquardt; momentum term; early termination; back propagation neural network; inversion prediction



Citation: Huang, J.; Luo, Y.; Shi, J.; Ma, X.; Li, Q.-Q.; Li, Y.-Y. Rapid Modeling of the Sound Speed Field in the South China Sea Based on a Comprehensive Optimal LM-BP Artificial Neural Network. *J. Mar. Sci. Eng.* **2021**, *9*, 488. <https://doi.org/10.3390/jmse9050488>

Academic Editor:
Charitha Pattiaratchi

Received: 1 April 2021
Accepted: 29 April 2021
Published: 1 May 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

The vertical structure of the sound speed in an ocean has a strong influence on sound propagation and determines underwater sound propagation characteristics. In practical applications, researchers must determine parameters that describe the seawater environment with depth to ensure the application’s accuracy. Additionally, knowledge of the sound speed information of a target sea area is essential when developing sounding, positioning, and underwater target detection and tracing [1,2]. However, due to the complexity of marine environments, the sound speed profile’s structure will change drastically with time and spatial position. Therefore, it is necessary to monitor underwater marine acoustic parameters continuously. Typically, seawater sound speed is calculated using a salinity–temperature depth system. Temperature, salinity, and depth are calculated using the

empirical formula of sound speed to determine the real sound speed [3,4]. However, this process is time-consuming and labor-intensive, and requires measurement of the sound speed profile in a larger area using a point-by-point method in real applications. Therefore, researchers have proposed the establishment of a model that uses historical data to predict the ocean's physical characteristics in a specific area [5,6].

Currently, models of sound speed profiles' structures can primarily be divided into two categories: analytical function models [7,8], such as the Munk model [7] and GDEM model [9]; and empirical orthogonal function (EOF) models [10,11]. The analytical function models describe a sound speed profile's structure using functional expressions and typically exhibit good scalability [12]. However, due to the function's limitations, the model structure is too ideal and cannot be used to describe a natural marine environment. Simultaneously, many parameters are used to construct the structure of a thermohaline surface; thus, analytical function models achieve low description efficiencies, and most parameters have no intuitive meaning. The empirical orthogonal function (EOF) model can achieve high-precision fitting of the sound speed profile using a reduced-order representation [11,13,14]; however, its shortcomings are equally marked. With high precision descriptions of the method, measurement data must be continuous in space. If the sampling data are measured at a discrete location, they must be interpolated to create equal-interval data. The EOF model is also affected by the sample depth of the sample, the calculated SSP cannot consist of points without data, and the approach cannot be described with high precision at points with missing data [15].

Therefore, to increase modeling accuracy, intelligent algorithms can be used to fit a nonlinear curve to calculate an accurate sound speed profile [16]. Neural networks can fit scattered data and perform classifications accurately. Zhang, Wenxiao et al. [17] trained and developed a BP neural network model to describe the complex nonlinear input and output characteristics of seawater salinity parameters, which are affected by many factors. The authors then used the artificial neural network inversion model to calculate the water depth inversion and achieved good results. Hu, H. et al. [18] used a small sample to verify that a BP artificial neural network predicts sound speed accurately for multi-beam sounding. Ai, R. et al. [19] used a BP artificial neural network's adaptive and self-learning capabilities [20] to train a model with historical data to create an inversion prediction model to perform inversion prediction of the sound speed profile. Hu, J. et al. [21] proposed a GA-optimized artificial neural network model that used ARGO data. Although computation time increased, fitting accuracy was markedly improved by that model.

Previous studies primarily used complex algorithms and low-precision data to establish prediction models. These methods cannot predict the sound speed profile well due to the limitation of the training data's accuracy. Simultaneously, due to the complexity of the algorithm, the training process is time-consuming. Therefore, based on existing research, this study uses high precision data (CTD observation dataset of scientific investigation over the South China Sea (2009–2012) [22]) and introduces the momentum term, normalization, the early termination method, and other optimization methods into the traditional BP artificial neural network model. This study also uses the LM algorithm to establish a fast sound field prediction model that meets the accuracy requirements of real applications. Finally, the model's application value is important. Jensen et al. [23] and Tindle et al. [24] proved that ray acoustics could effectively simulate the work of acoustics devices in an underwater environment. So, in this paper, the model's sound speed profile's accuracy and application value are confirmed using sound ray tracing. In practical applications, the proposed model provides important scientific significance and reference value.

2. Data and Experimental Platform

2.1. Simulation Platform

This study uses MATLAB (2020a) to calculate all simulations that construct the sound speed field in the South China Sea. The experimental hardware included an Intel Core i7-9750H CPU with 6 cores, 12 threads, and 16 GB of RAM.

2.2. Data Sources

The CTD observation dataset of scientific investigation over the South China Sea (2009–2012) was used as the sample space in this study [22]. The dataset included data from 18° N, 10° N, 6° N sections and 113° E meridian sections with 75 stations, in which the deepest depth was 1550 m. From 2009 to 2012, a comprehensive survey of four voyages in different seasons was completed. Usually, each station has 769 points of data in a voyage. In this paper, the total data set consisted of around 173,025 points, and each point included the point's spatial coordinate and sound speed. Compared with other CTD data sets (e.g., the Levitus database and the global CTD grid data), this paper's dataset has more advantages, which are shown below.

1. The data accuracy has a higher spatial resolution than other datasets. For example, the data's sampling interval is every 2 m in vertical profile, compared with the Levitus database, which has a 10 m interval in the sea surface and 100 m at other depths. That means these data will describe an inaccurate sound speed profile and negatively affect the model's training.
2. The data have higher precision than other datasets. The data set is contributed through scientific investigation. Compared with other datasets, the in-site measured data have higher precision, whether in terms of the measuring instrument or the data's quality control. The training data's precision will have a significant effect on the model.
3. The data set has enough data in the model's training. The 75 stations, in three years, provided around 173,025 rows of data, which obviously satisfy three layers of artificial neural network used in this model.

Moreover, the important technical performance indicators are shown below.

1. Specific conductance: 0 ~ 7 S/m with an accuracy of 0.0003 S/m;
2. Temperature: -5 ~ +35 °C with an accuracy of 0.001 °C;
3. Pressure: 0 ~ 10,000 Pa with an accuracy of 0.015% FS (full-scale), equal to an accuracy of 1.5 Pa.

The point distribution is shown in Figure 1.

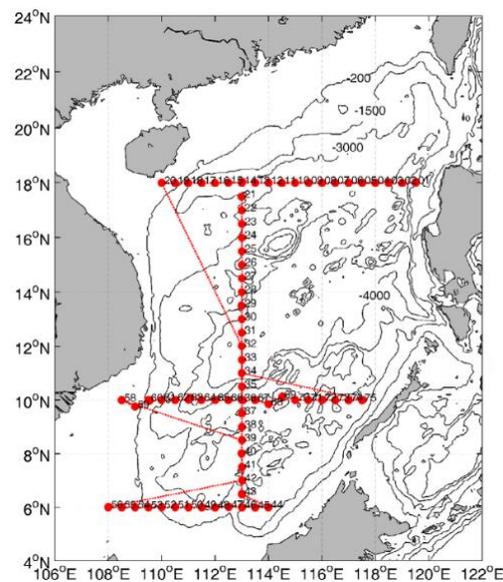


Figure 1. Sites of the dataset in the South China Sea.

Each dataset contains 11 types of information, which are listed in Table 1.

Table 1. Dataset content.

Number	Name	Usage and Meaning	Unit	Sample
1	Site Name	Site name, format: kj01–kj75	-	kj01
2	Date	Measurement date, format: yyyyymmdd	-	20,090,617
3	Time	Measurement time, format: hhmmss	-	145,446
4	Latitude	Latitude at data measurement	°N	18.001
5	Longitude	Longitude at data measurement	°E	119.502
6	Depth	Site water depth	m	3008.0
7	Layer measurement	Data observation layer depth	m	1.0
8	Temperature	Temperature value	°C	28.72
9	Salinity	Salinity value	PSU	33.4923
10	Density	Sigma-theta ($\sigma\text{-}\theta$) density value	kg/m ³	21.024
11	Sound speed	Sound speed value	m/s	1541.3

2.3. Data Selection

To ensure the authenticity of training, it is necessary to select data for the training and verification sets randomly. In this study, 80% of the data were randomly selected as the training set, and 20% were used as the verification set. With the early termination method, 80% of the training set is divided into 70% for training and 10% for the confirmation sample. Figure 2 shows the classification of the selected points at a training instance.

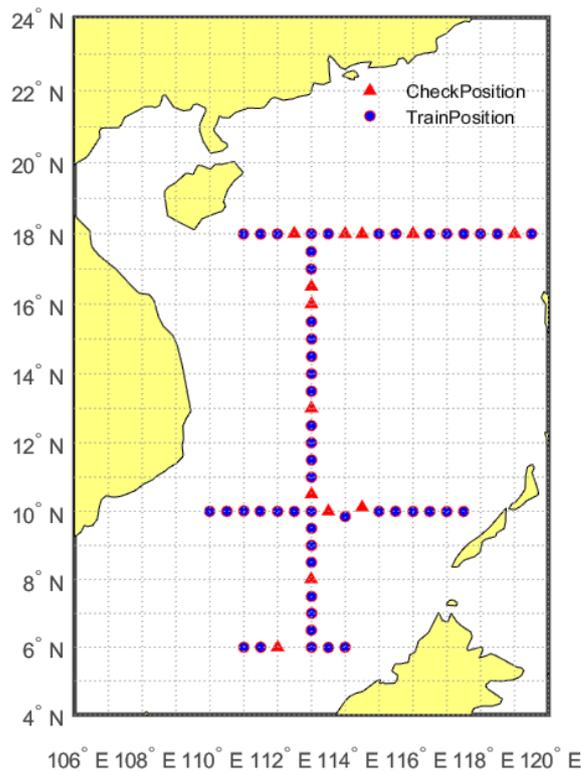


Figure 2. Classification of selected sites.

3. Methodology

A BP artificial neural network is a typical neural network algorithm that uses the error after the output to estimate the error of the direct leading layer of the output layer [25,26]. The BP neural network then uses this error to estimate the previous layer’s error, and passes it back, layer by layer, to obtain the error estimates of all layers [27].

Figure 3 shows the overall process flowchart of the following processes. First, we divide the measured data into training and checking sets. The training sets are imported

into the model. Then, the network parameters are initialized, including connection weights and bias values. Typically, the initial parameters are calculated using random numbers. Second, we calculate the neural layers' output. In this section, we compare various activation functions. Third, the output error is calculated based on the input data to adjust the connection weights. The changing range can be optimized with the momentum term, which can effectively adjust the connection weight and avoid being stuck in a local minima. Then, we check the network accuracy: if the output error meets the required precision, the model will be saved; otherwise, we check the iterations. If the training epochs reach the training times, the model will be saved and exported to successive epochs. Between these two judgments, the paper adds an early termination as an optimal method to mitigate overfitting.

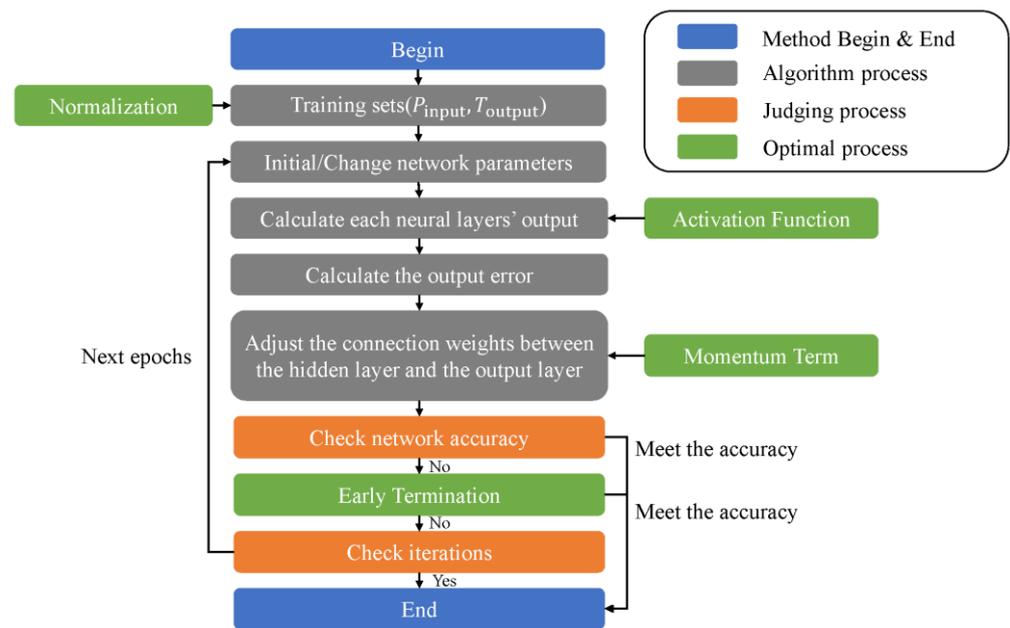


Figure 3. Flowchart of the methodology of the back propagation neural network.

Each part of the methodology in Figure 3 is described in detail in the following subsections. After that, a presentation of the sound ray trace method has been given. Typically, we use ray acoustics to manage sound propagation. This study simulates sound ray traces with measured sound speed profiles and model sound speed profiles. Then, this study explores the feasibility of applying this sound speed prediction model using this method. The section's structure is shown below.

Figure 4 shows the structure of this section. First, in Section 3.1, Data Preprocessing, this paper introduces the data preprocessing, including the sample divisions, and the meaning of the sample matrix P_{input} and the target vector T_{output} . Then, in Section 3.2, BP Network Design, the article introduces the network design, including the hidden layer, input and output layer, and activation functions. This section is the second process of machine learning in practical usage. Third, the paper introduces a basic training algorithm in Section 3.3, BP Network Training Algorithm. Next, Section 3.4, BP Model Optimization, is the critical part of this section. In this section, the article brings the optimization in the model. The optimization consists of selecting the best training function (LM algorithm), appending the momentum term, and using normalization and early termination. Finally, the sound ray trace simulation is introduced in Section 3.5. This method is used in the evaluation of the model's practical performance. In this section, the paper introduces the theory of sound ray trace and its parameters' calculation. Each section is shown below.

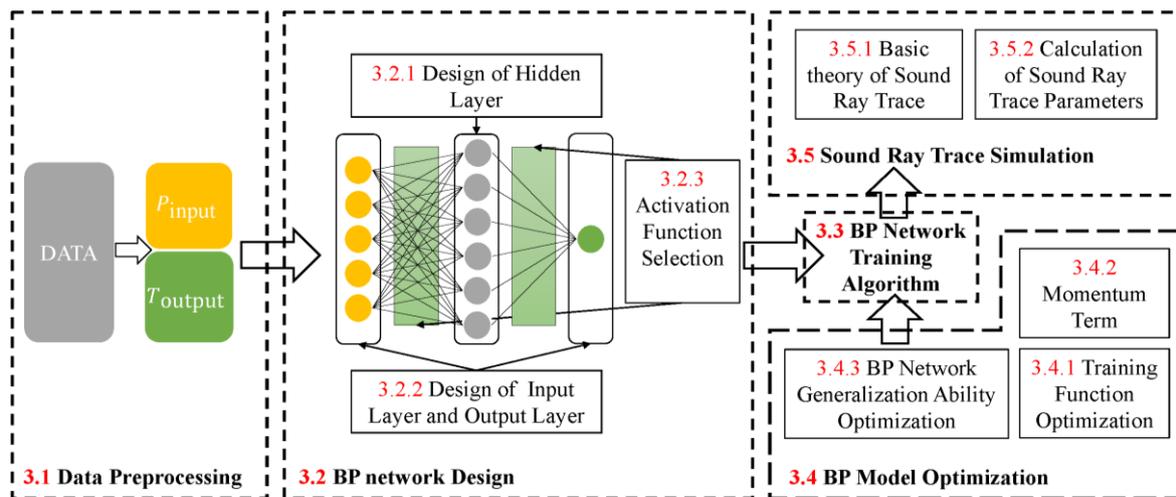


Figure 4. The structure of the methodology.

3.1. Data Preprocessing

The data must be divided into the sample matrix P_{input} and the target vector T_{output} , the structure of which is as follows:

$$P_{input} = \begin{bmatrix} r_{date}^1 & r_{time}^1 & r_{lat}^1 & r_{lng}^1 & r_d^1 \\ r_{date}^2 & \ddots & \vdots & \vdots & r_d^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{date}^n & r_{time}^n & r_{lat}^n & r_{lng}^n & r_d^n \end{bmatrix} \quad (1)$$

$$T_{output} = \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} \quad (2)$$

where r_{date} indicates the date (year, month, day) in “doy” (day of year) format; r_{time} indicates the result, which converts the time (hours, minutes, and seconds) into the number of hours starting from 00:00 of the day; r_{lat} indicates the latitude of the point; r_{lng} indicates the point’s longitude; r_d is the measurement depth of the sampling point. In Equation (2), s_i represents the sound speed data corresponding to the i row of data. Each row of T_{output} corresponds to that of P_{input} . In other words, taking the first row as an example, the input data with five elements is the sample matrix, and the target vector is the sound speed corresponding with the input data. If we combine the sample matrix and target vector as a matrix $[P_{input}, T_{output}]$, this matrix is the complete data of the data set.

Considering the target sample matrix P_{input} as an example, a $P_{N \times 5}$ matrix is formed after normalization, which indicates that the matrix’s data have been converted into the range between zero and one, where N is the total amount of data in the training set. Due to the strong fitting ability of the neural network, it is necessary to randomize the data sequence to prevent these nontarget features, such as the regular arrangement of the input data, from being learned, to ensure the convergence and generalizability of the results. The neural network used in this study uses the gradient descent rule to achieve supervised learning; thus, randomization is critical. Without randomization, the training results have difficulty converging to offset values. Only by ensuring the randomness of data can the algorithm’s training results converge as much as possible. Therefore, P_{input}^i s rows are not consistent with regular data, which indicates that the first row may be the first site’s data on the sea surface, and the second row may be the other site’s data at random depths.

3.2. BP Network Design

BP network design, including the input layer, hidden layer, and output layer’s design, is the second step of deep learning. After ensuring the layers’ design, the model can be trained. The neural network is a three-layer BP neural network model that includes an input layer, a hidden layer, and an output layer. The three-layer BP neural network model used in this article is shown in Figure 5.

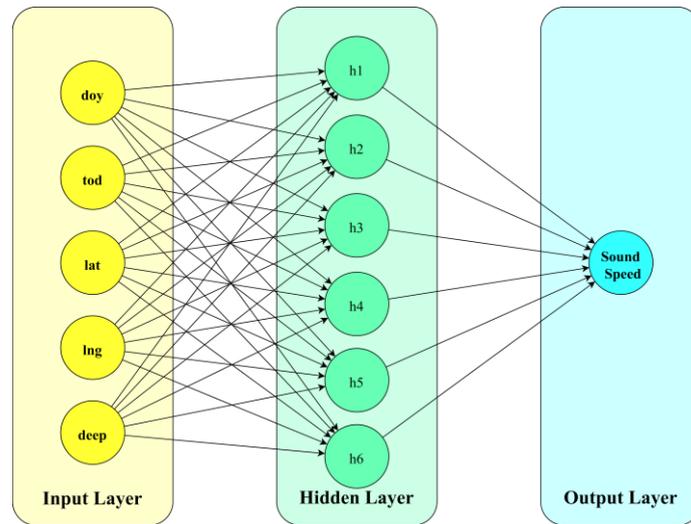


Figure 5. Three-layer BP neural network model (doy: day of year; tod: time of day).

In this model, the input layer consists of 5 elements: the sample sites’ latitude, longitude, depth, and sampling time, which includes the day of the year (doy) and the time of day (tod). The model’s output is the sample sites’ sound speed. In other words, the model’s input is one point’s spatial coordinates, and the output is the point’s sound speed. Taking 75 sets as an example, in Figure 6, the input is 75 positions’ five elements, and the output is the sound speed profile at these sets. Normally, the CTD data consists of 769 points (measured every 2 m). Therefore, if we input 769 points into this model (just changing the input data’s depth), the output result constitutes the sound speed profile.

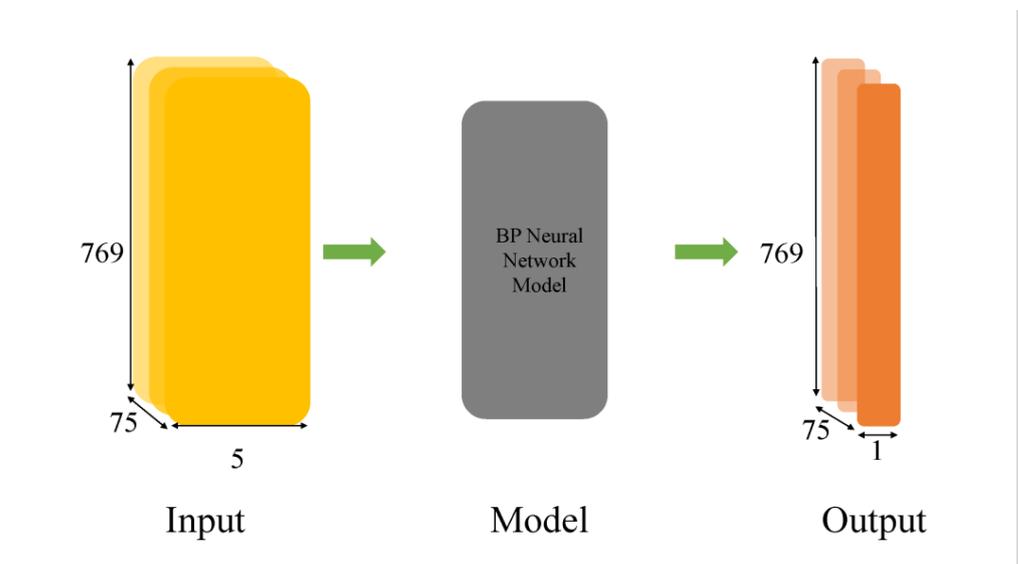


Figure 6. Input and output of the BP neural network model.

3.2.1. Design of Hidden Layer

First, the hidden layer's number determination is the critical work of the model. In a BP network, any continuous function on the closed interval can be approximated by a single hidden layer BP network so that a three-layer BP network can complete any n-dimensional to m-dimensional mapping [28]. In practical applications, the number of hidden neurons is causally related to the problem's requirements. Too many hidden neurons will lead to more learning time, more error, and less fault tolerance. Besides this, too many hidden neurons will bring an overcautious judgment to the model that fails to recognize samples that have not been seen before. This phenomenon is usually called overfitting. However, a lack of hidden neurons usually causes underfitting. Thus, there must be an optimal number of hidden neurons.

In actual experiments, empirical formulas, which are different approaches to estimating the number of hidden neurons, are used to select the optimal number of hidden neurons:

1. Empirical Formula (1) [29]:

$$\sum_{i=0}^{N_{input}} C_{N_{hid}}^i > k \tag{3}$$

where k is the number of samples, N_{hid} is the number of hidden neurons, N_{input} is the number of input neurons, and i is a constant in the range of $[0, N_{input}]$.

2. Empirical Formula (2) [29]:

$$N_{hid} = \sqrt{N_{input} + N_{output} + a} \tag{4}$$

where N_{input} is the number of input neurons, N_{output} is the number of input neurons, and a is a constant in the range of $[1, 10]$.

3. Empirical Formula (3) [29]:

$$N_{hid} = \frac{N_{input} + (N_{output}, N_{class})_{max}}{2} \tag{5}$$

where N_{output} is the number of output neurons; N_{input} is the dimension of the input vector, which is equal to the number of input neurons; and N_{class} is the number of target classifications. In this study, the question is a regression problem, not a classification problem; thus, the variable N_{class} is equal to zero, and $(N_{output}, N_{class})_{max}$ is the maximum of N_{output} and N_{class} .

4. Empirical Formula (4) [30]:

$$N_{hid} = \log_2 N_{input} \tag{6}$$

where N_{input} is the number of input neurons.

5. Empirical Formula (5) [30]:

$$N_{hid} = 1 + [N_{input} \times (N_{output} + 2)] \times 0.5 \tag{7}$$

where N_{input} is the number of input neurons, and N_{output} is the number of output neurons.

6. Empirical Formula (6) [31]:

$$N_{hid} = 2N_{input} + 1 \tag{8}$$

where N_{input} is the number of input neurons

7. Empirical Formula (7) [32]:

$$N_{hid} \leq \frac{N_{train}}{R \times (N_{input} + N_{output})} \tag{9}$$

where N_{input} is the number of input neurons, N_{output} is the number of output neurons, and N_{train} is the number of samples in the training set.

Based on Equations (3)–(9), the optimal number of hidden neurons can be determined to be in the range of [2, 12]. Then, the gradient descent method is applied. First, more neurons are selected, and then, the number of hidden neurons is gradually decreased until it reaches a reasonable number of hidden neurons.

Figure 7 shows the results of hyperparameter training. The average root mean square error (RMSE) and training time are compared for different numbers of hidden units. The result shows that using too many neurons in the hidden layers can result in several problems.

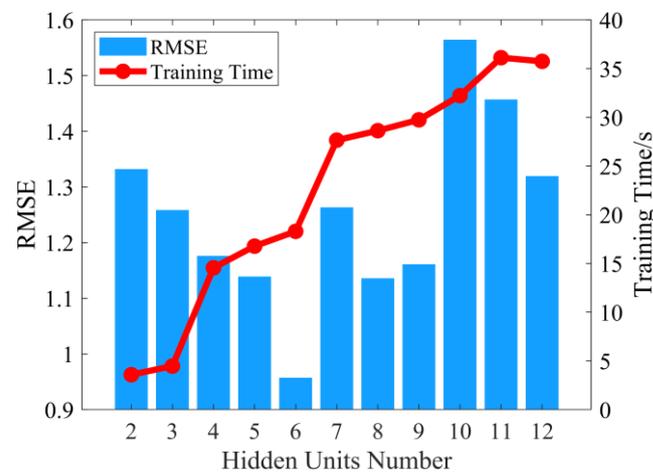


Figure 7. Hyperparameter training results of different hidden neuron numbers.

First, too many neurons in the hidden layers may result in overfitting, which occurs when the neural network has so much information processing capacity that the limited amount of information in the training set is not sufficient to train all of the neurons in the hidden layers. A second problem can occur even when sufficient training data are available. An inordinately large number of neurons in the hidden layers can increase the time it takes to train the network. The amount of training time can increase to the point that it is impossible to train the neural network adequately.

Second, using too few neurons in the hidden layers will result in underfitting, which occurs when there are too few neurons in the hidden layers to adequately detect the signals in a complex data set.

Therefore, a compromise must be reached between too many and too few neurons in the hidden layers, and the selection of an architecture for the neural network requires trial and error. Equations (3)–(9) provide a starting point or range for us to consider. When keeping all other conditions the same (including the training data, training function, etc.), the model’s accuracy and computation time are optimal when the number of hidden layers is 6.

3.2.2. Design of Input Layer and Output Layer

Following Section 3.1 (Data Processing), we use the date, time, longitude, latitude, measurement layer’s depth, and sound speed in the sample space as the training set. We then construct the target sample P_{input} and target vector T_{output} with n rows, 5 columns and n rows, and 1 column, in Equations (1) and (2).

Because the input matrix is a 5-column matrix, the number of neurons in the input layer is selected as the number of columns in the input matrix. The output result is a sound speed. The number of neurons in the output layer is 1.

3.2.3. Activation Function Selection

Because linear models have a severe limitation, a complex curve should be introduced to the neural network to achieve a more flexible model. Piecewise linear curves could be used to approximate the continuous and complex curve. In this paper, the activation function is an approximation of the piecewise linear curve. In other words, the neural network’s activation function adds nonlinear factors to the network to solve the problems that the linear model cannot solve. If the activation function is not used, no matter how many layers the neural network has, the final output is a linear combination of inputs, equivalent to having no hidden layer. Therefore, using a nonlinear function as the activation function can allow the neural network to approximate any function.

The BP neural network used in this article requires the activation function to be derivable. The reason is shown below. After picking an initial value of the network’s gradient descent process, the model will upgrade the parameter to find the loss function’s global minima. The activation function’s differential will guide the function’s decline. In other words, when adjusting the parameter using the loss function, the parameter is often calculated using the gradient descent method, and it is necessary to obtain the partial derivative of the adjusted parameter. Therefore, the sigmoid function and the tan function are suitable for use with BP neural networks. The two functions and their derivatives are shown in Figure 8.

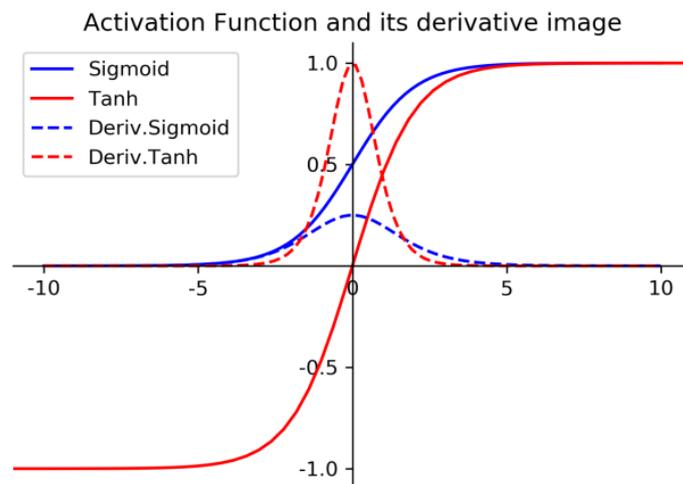


Figure 8. Curve of the Sigmoid and Tanh functions.

1. Sigmoid

The sigmoid function is commonly used in machine learning to map any input value onto the output range (0, 1):

$$sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{10}$$

The advantage of this function is that the function is nonlinear and separate from the binary output. The sigmoid function can output any value between (0, 1) and can be used to represent probability. The output value of the final function is within a given range and will not output infinity. However, the sigmoid function has shortcomings. First, when the input value z approaches $\pm\infty$, the output value approaches 0 or 1. In this case, the corresponding gradient is small or even disappears. The network will also not be learning, particularly when using the gradient descent algorithm. Second, in actual operation, the exponential operation overhead is too high.

2. Tanh

The Tanh function is also commonly used in machine learning. It can map any input value onto the output range $(-1, 1)$:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{11}$$

This activation function is similar to the sigmoid function, except that the output range is $(-1, 1)$ and, thus, has the same advantages and disadvantages as the sigmoid function. However, the tanh function performs better than the sigmoid function when used as a hidden layer activation function for the following reasons. First, compared to the sigmoid function, the tanh function’s average value is 0, which has a good effect on data centering. Second, the tanh function has a large gradient near 0, which allows for fast convergence. Therefore, the tanh function is used as the activation function in this article.

3.3. BP Network Training Algorithm

This sections’ total important symbol meanings are shown in Table 2.

Table 2. Meanings of symbols in the BP Network Training Algorithm.

Name	Symbol
Network input	$P_{\text{input}} = (r_1, r_2, \dots, r_n)$
Network target vector	$T_{\text{input}} = (s_1, s_2, \dots, s_q)$
Intermediate unit input vector/output vector	$S_k = (s_1, s_2, \dots, s_p), B_k = (b_1, b_2, \dots, b_p)$
Output layer unit input vector/output vector	$L_k = (l_1, l_2, \dots, l_q), C_k = (c_1, c_2, \dots, c_q)$
Connection right from the input layer to the middle layer	$w_{ij}(i = 1, 2, \dots, n; j = 1, 2, \dots, p)$
Right to connect the middle layer to the output layer	$v_{jt}(j = 1, 2, \dots, p; t = 1, 2, \dots, q)$
Output threshold of each unit in the middle layer	$\theta_j(j = 1, 2, \dots, p)$
Output threshold of each unit of the input layer	$\gamma_j(j = 1, 2, \dots, p)$

The basic steps of the BP neural network algorithm are shown below:

- Step 1: Provide each connection weight w_{ij} , v_{jt} , threshold θ_j and γ_j a random value in the interval $(-1, 1)$.
- Step 2: Randomly select a set of input and target samples $P_{\text{input}}^k = (r_1^k, r_2^k, \dots, r_n^k)$, $T_{\text{input}}^k = (s_1^k, s_2^k, \dots, s_q^k)$ and input it to the network.
- Step 3: In Equation (12), use the input sample $P_{\text{input}}^k = (r_1^k, r_2^k, \dots, r_n^k)$, connection weight w_{ij} and threshold θ_j to calculate the input s_j of each unit in the middle layer, and then in Equation (13), use s_j to calculate the value of each unit in the middle layer through the transfer function Output b_j :

$$s_j = \sum_{i=1}^n w_{ij}a_i - \theta_j, j = 1, 2, \dots, p \tag{12}$$

$$b_j = f(s_j), j = 1, 2, \dots, p \tag{13}$$

- Step 4: In Equation (14), use the input b_j of the intermediate layer, the connection weight v_{jt} and the threshold γ_t to calculate the output L_t of each unit of the output layer. Then, calculate the corresponding C_t of each unit of the output layer using the transfer function in Equation (15):

$$L_t = \sum_{j=1}^p v_{jt}b_j - \gamma_t, t = 1, 2, \dots, q \tag{14}$$

$$C_t = f(L_t), t = 1, 2, \dots, q \tag{15}$$

- Step 5: Use the network target vector $\mathbf{T}_k = (y_1^k, y_2^k, \dots, y_q^k)$ and the actual output of the network \mathbf{C}_t to calculate the generalized error d_t^k of each unit of the output layer in Equation (16):

$$d_t^k = (y_t^k - C_t) \cdot C_t(1 - C_t), t = 1, 2, \dots, q \tag{16}$$

- Step 6: In Equation (17), use the connection weight v_{jt} , the generalized error d_t of the output layer and the output b_j of the intermediate layer to calculate the generalized error e_j^k of each unit of the intermediate layer:

$$e_j^k = \left[\sum_{i=1}^q d_i \cdot v_{jt} \right] b_j(1 - b_j) \tag{17}$$

- Step 7: In Equation (18), use the generalized error d_i^k of each unit of the input layer and the output b_j of each unit of the intermediate layer to modify the connection weight v_{jt} and the threshold γ_j :

$$\begin{aligned} v_{jt}(N + 1) &= v_{jt}(N) + \alpha \cdot d_i^k \cdot b_j \\ \gamma_j(N + 1) &= \gamma_j(N) + \alpha \cdot d_t^k \\ t = 1, 2, \dots, q; j &= 1, 2, \dots, p; 0 < \alpha < 1 \end{aligned} \tag{18}$$

- Step 8: In Equation (19), use the generalized error e_j^k of each unit of the middle layer and the input of each unit of the input layer $\mathbf{P}_k = (a_1, a_2, \dots, a_n)$ to modify the connection weight w_{ij} and the threshold θ_j :

$$\begin{aligned} w_{ij}(N + 1) &= w_{ij}(N) + \beta e_j^k a_i^k \\ \theta_j(N + 1) &= \theta_j(N) + \beta e_j^k \\ i = 1, 2, \dots, n; j &= 1, 2, \dots, p; 0 < \beta < 1 \end{aligned} \tag{19}$$

- Step 9: Randomly select the next learning sample vector and provide it to the network. Return to Step 3 until the training of m training samples is completed.
- Step 10: Reselect a set of input and target samples randomly from the m of learning samples, and return to Step 3 until the network global error E is below a preset minimum value; the network thus converges. If the learning time is greater than the present value, the network cannot converge.

3.4. BP Model Optimization

In this section, this paper will introduce the optimization of the training function, momentum term, normalization, and early termination. These methods are used to optimize the bp-training algorithm.

3.4.1. Training Function Optimization

In this section, the paper will briefly introduce the four different training functions. First, the paper compares the traditional algorithm’s training function, the Newton method, and the LM algorithm. Then, it has a brief introduction to these algorithms, and an analysis of their mathematical differences. At the end of this section, four functions’ training results are given. The result shows that the LM algorithm has a significant advantage in convergence speed.

When training the BP neural network, data fitting, parameter estimation, and function approximation are often required, and these training processes can typically be attributed to the processing of nonlinear least-squares problems in Equation (20):

$$\min_{x \in R^n} E(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2} \sum_{i=1}^m [r_i(x)]^2, m \geq n \tag{20}$$

The traditional BP algorithm updates the parameters using the fastest gradient descent method, which is along the opposite direction of the gradient. The parameters are updated according to specific step sizes. Although the evaluation function (loss function)—the difference between the expected output and the final layer’s output—reaches a minimum value, this method ignores the second derivative term. The second derivative term can be used to adjust the convergence rate. Taking the second derivative term into consideration, the model’s convergence rate will have an apparent improvement. When the evaluation function is small, the convergence is linearly convergent and converges slowly; therefore, it is often used as the method in the initial stage of optimization. While the Newton method uses the second derivative, the final stage exhibits a fast convergence speed and good convergence, but is, thus, not suitable for the initial stage. Simultaneously, the Newton method must be used to calculate the Hessian matrix, which is the second derivative information for each step; unfortunately, this calculation is complex.

Figure 9 is the summary of the above paragraphs. In this figure, the traditional bp algorithm (gradient descent method) shows a rapid convergence rate in the initial stage but shows a slow and poor performance in the final stage. The newton method shows a good and rapid convergence in the final stage but does not have a good effect on the initial stage. The LM algorithm combines both methods’ advantages, showing a great performance in all stages. Based on this situation, we use the Levenberg–Marquardt algorithm in BP and compare it to other methods, and associated analyses and comparisons are shown below.

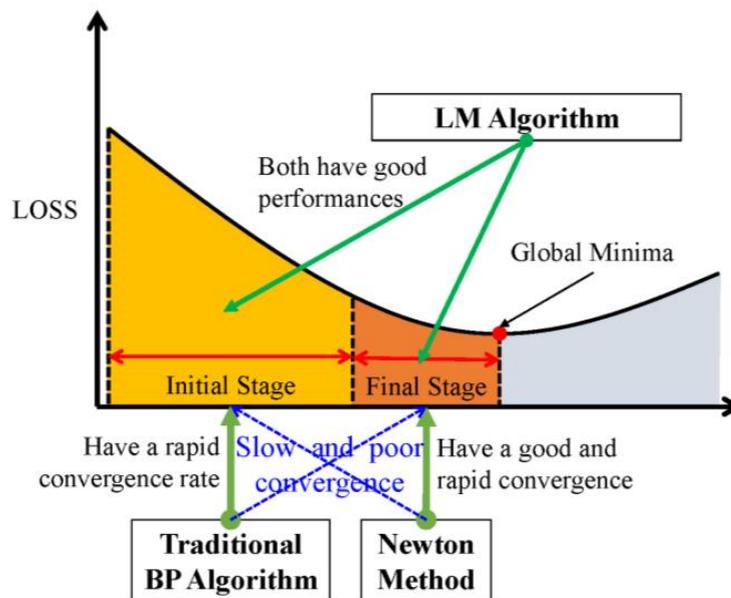


Figure 9. Three different algorithms’ performance in the loss function.

The LM algorithm is a fast algorithm that uses standard numerical optimization techniques, which solve the unconstrained optimization problem and the optimization problem with constraints by iteratively approaching the optimal solution gradually, respectively. Furthermore, the LM algorithm is a combination of the gradient descent method and the Gauss–Newton method. The LM algorithm achieves the local convergence provided by the Gauss–Newton method and exhibits the global characteristics of the gradient descent method. In the loss function, the local convergence means the Gauss–Newton method’s rapid convergence, and the global characteristics mean the advantage in the initial stage. In practical applications, the characteristics of the approximate second-order derivative information near the extreme points are used to approximate the quadraticity to speed up optimization and convergence. The proposed algorithm is as follows:

Let $x^{(k)}$ denote the vector of the k iterations, and the newly formed vector $x^{(k+1)}$ can be obtained according to Equation (21):

$$x^{(k+1)} = x^{(k)} + \Delta x \tag{21}$$

For Newton’s method:

$$\Delta x = -[\nabla^2 E(x)]^{-1} \nabla E(x) \tag{22}$$

where $\nabla^2 E(x)$ is the Hessian matrix of the error-index function $E(x)$, and $\nabla E(x)$ is the gradient. Supposing the error-index function $E(x)$ is:

$$E(x) = \frac{1}{2} \sum_{i=1}^N e_i^2(x) \tag{23}$$

where $e(x)$ is the error, then:

$$\begin{aligned} \nabla E(x) &= J^T(x)e(x) \\ \nabla^2 E(x) &= J^T(x)e(x) + S(x) \end{aligned} \tag{24}$$

where $S(x) = \sum_{i=1}^N e_i(x)\nabla^2 e_i(x)$; and $J(x)$ is the Jacobian matrix, which is defined in Equation (25):

$$J(x) = \begin{bmatrix} \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \dots & \frac{\partial e_1(x)}{\partial x_n} \\ \frac{\partial e_2(x)}{\partial x_1} & \frac{\partial e_2(x)}{\partial x_2} & \dots & \frac{\partial e_2(x)}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial e_N(x)}{\partial x_1} & \frac{\partial e_N(x)}{\partial x_2} & \dots & \frac{\partial e_N(x)}{\partial x_n} \end{bmatrix} \tag{25}$$

The calculation rules for the Gauss–Newton method are based on Equation (26):

$$\Delta x = -[JT(x)J(x)]^{-1}J(x)e(x) \tag{26}$$

The form of the LM algorithm is Equation (27):

$$\Delta x = -[JT(x)J(x) + \mu I]^{-1}J(x)e(x) \tag{27}$$

where the coefficient $\mu > 0$ is a constant, and I is the identity matrix.

Equation (26) shows that if $\mu = 0$, then the Gauss–Newton method is used; if μ is large, the LM algorithm is similar to the gradient descent method. For each iteration step, μ decreases and approaches the error target, as in the Gauss–Newton method. When the Gauss–Newton method approaches the minimum error, the calculation speed increases and the accuracy exceeds that of other methods. Because the LM algorithm uses an approximate second-order derivative, it converges much more quickly (up to 10x faster) than the gradient descent method. Additionally, because $|J^T(x)J(x) + \mu I|$ is definitely positive, there must be a solution. In the Gauss–Newton method, $|J^T(x)J(x)|$ is either full or does not remain under discussion.

In a real application, μ is a tentative parameter. For a given parameter, if the obtained error-index function $E(x)$ decreases, then μ decreases; otherwise, μ increases. The computational complexity of the LM algorithm is $O(n^3/6)$. If n is large, the number of required calculations and amount of storage are large. However, improving the iteration efficiency can significantly improve overall performance, particularly when accuracy is essential.

In Figure 10, four training functions are used to train the model with part of the data. After 1000 epochs, the error graph shows that the LM algorithm achieves faster convergence speeds compared to other algorithms.

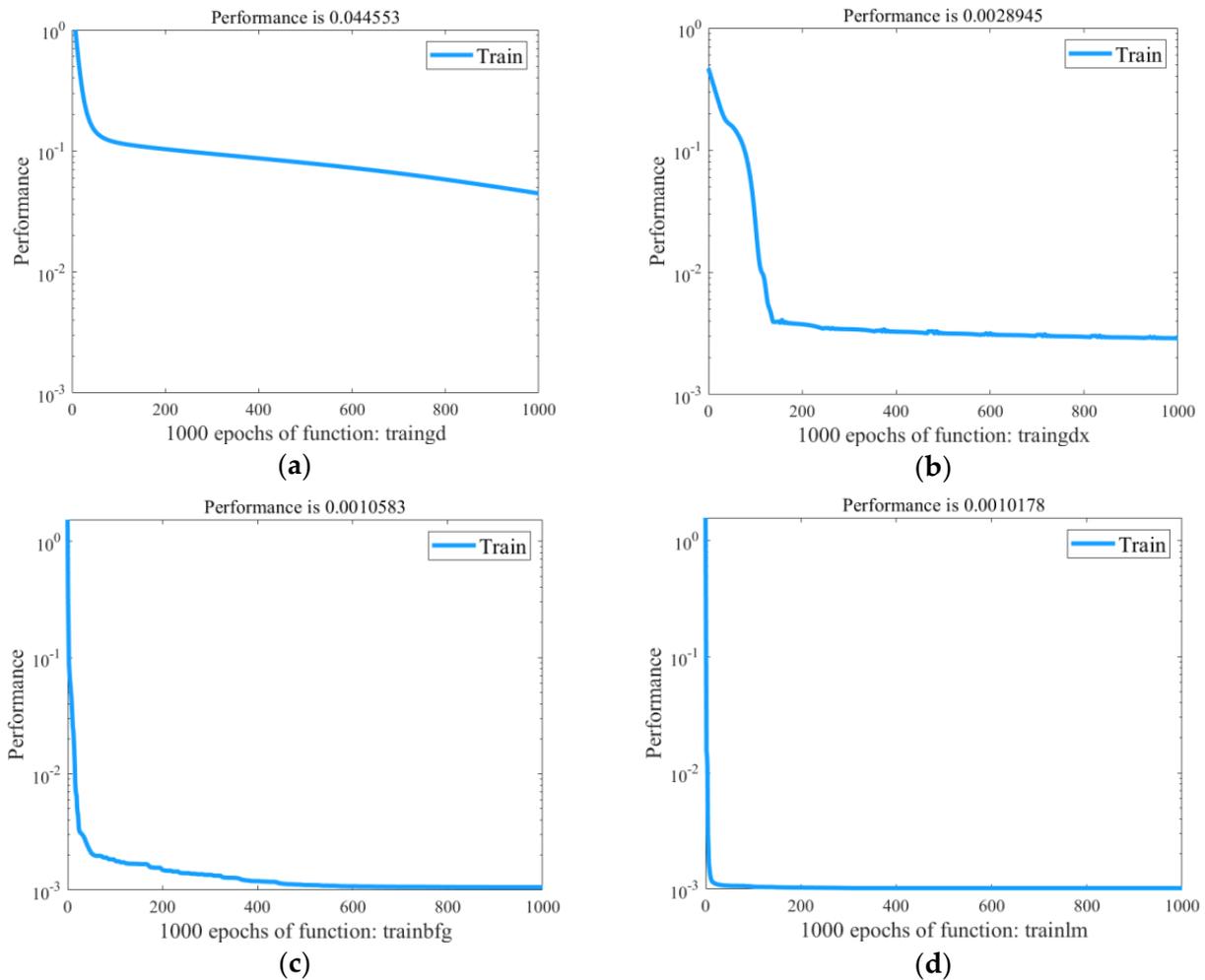


Figure 10. Performances of four training functions: (a) Performance of gradient descent backpropagation; (b) performance of gradient descent with momentum and adaptive learning rate backpropagation; (c) performance of BFGS quasi-Newton backpropagation; and (d) performance of Levenberg-Marquardt backpropagation.

3.4.2. Momentum Term

In the BP algorithm, the learning step size η is related to the speed and stability of the network convergence. If η is too large, network instability will occur if the network convergence speed is increased too much. If η is too small, this phenomenon can be avoided; however, the convergence speed will decrease. To resolve this conflict, a momentum term is introduced in this study to fine-tune the weight using the following equation:

$$\Delta w_{ij}(n) = \alpha \Delta w_{ij}(n - 1) + \eta \delta_j(t) v_i(n) \tag{28}$$

where α is the momentum term and is typically a positive number.

Rewriting these formulae as time series with training time t as a variable, t is within $(0, n)$, and the formula can be considered to be $\Delta w_{ij}(n)$ as the first-order difference equation:

$$\Delta w_{ij}(n) = \eta \sum_{i=0}^n \alpha^{n-i} \delta_j(t) v_i(n) \tag{29}$$

where $\delta_j(t) v_i(n)$ can be calculated by:

$$\delta_j(t) v_i(n) = - \frac{\partial E(n)}{\partial w_{ij}(n)} \tag{30}$$

which can be rewritten as:

$$\Delta w_{ij}(n) = -\eta \sum_{t=0}^n \alpha^{n-t} \frac{\partial E(n)}{\partial w_{ij}(n)} \tag{31}$$

Therefore, in real applications, the correction amount $\Delta w_{ij}(n)$ is the sum of a series of weighted exponential sequences. When the momentum constant satisfies $|\alpha| \in [0, 1)$, the sequence converges. When $\alpha = 0$, there is no momentum term. In theory, α can be either positive or negative; however, negative values are used in practical applications.

The adjustment speed control is related to $\frac{\partial E(n)}{\partial w_{ij}(n)}$. When $\frac{\partial E(n)}{\partial w_{ij}(n)}$ has the same sign as the previous time, its weighted summation of the value increases to make $\Delta w_{ij}(n)$ larger, thereby increasing the adjustment speed of w during stable adjustment. Similarly, when $\frac{\partial E(n)}{\partial w_{ij}(n)}$ is negative, the weighted summation result decreases, which has a stabilizing effect.

As shown in Figure 11, after training 10 times, the blue bars represent the RMSE of the model without momentum, the green bars are the model's momentum terms equal to 0.9, and the red line shows the improvement when we bring the momentum terms into the model. Significantly, the introduction of momentum terms during training can effectively improve training accuracy.

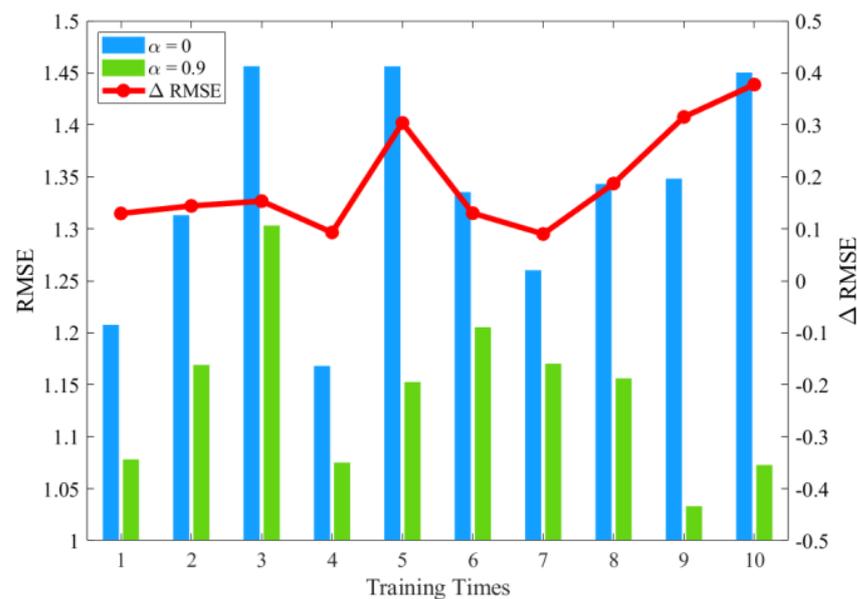


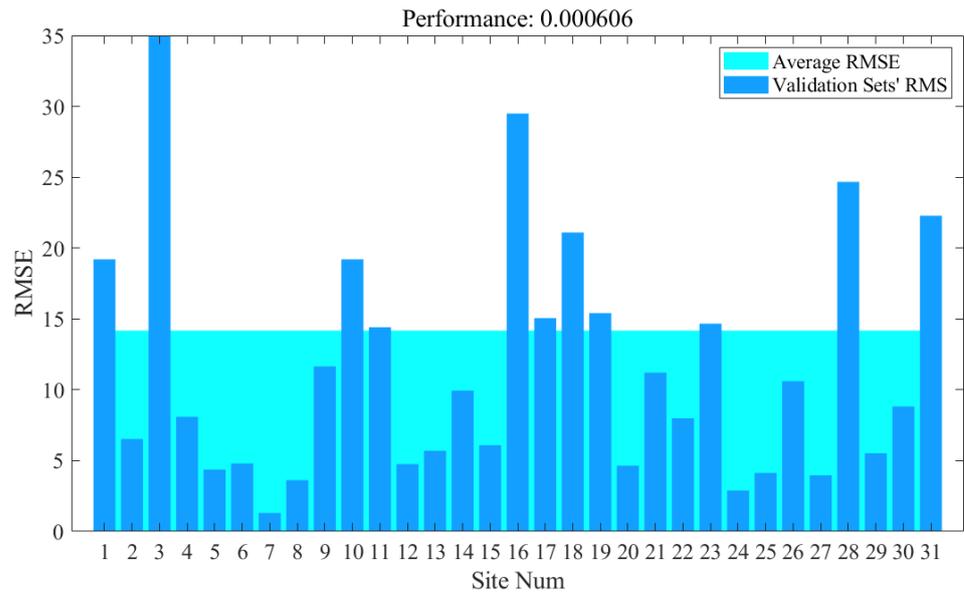
Figure 11. Performance of network with and without the momentum term.

3.4.3. BP Network Generalization Ability Optimization

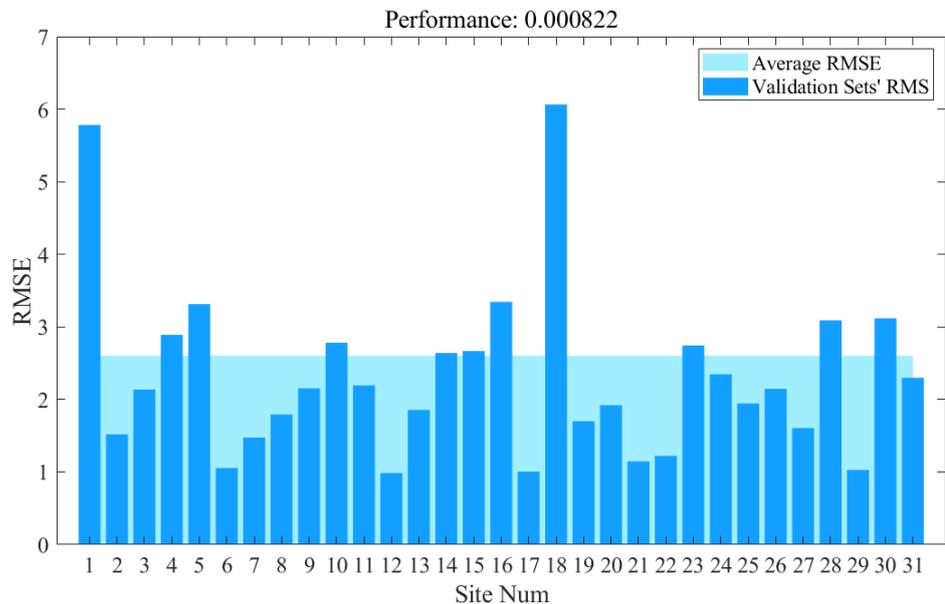
In this section, the paper first includes a brief introduction to overfitting. In practical usage, overfitting is common. To reduce this phenomenon, the scientist usually uses some of the methods below. First, selecting more training data; second, using data augmentation, which is based on the questions' comprehension to create more useful data. Third, choosing the less flexible model, which is the model with fewer neurons or limited hidden layers. These methods have been used or been limited in this paper's model. Therefore, this paper selected the last one, using the normalization and early termination (early stopping), which have been introduced in this paper.

When training neural networks, overfitting is one of the most common phenomena. The error of the training set samples can be small; however, the error for the new sample data outside the training set will be large, which shows that the network has memorized trained samples but cannot generalize to new samples. The following figures show an example of overfitting.

In Figure 12, the new sample error increases rapidly as the sampling error decreases in the case of overtraining. An effective way to solve this problem is to adjust the network’s size to make it sufficiently fit the data: the larger the network scale is, the stronger the function mapping function of the network will be. Therefore, if the network scale is sufficiently small, overfitting can be eliminated. This part of the study is performed when the optimal hidden layer number is selected. Additionally, there are two methods to improve the network’s generalizability: the normalization method and the early termination (early stopping) method.



(a)



(b)

Figure 12. Example of overfitting: (a) the model’s RMSE when the performance index is 0.000606; and (b) the model’s RMSE when the performance index is 0.000822.

These two methods are described like this:

1. Normalization Method

The data normalization method is a standard processing method for data in neural networks. Data normalization converts all data into a number in the range of [0, 1]. This method can cancel out the difference in the order of magnitude between each dimension's data and mitigate the significant difference in the magnitude of the input/output data, which could mitigate significant network prediction errors.

The basic idea of the normalization method is to correct the network error performance parameters. The typical error performance function is the mean square error function in Equation (32):

$$E = mse = \frac{1}{N} \sum_{i=1}^N (e_i)^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2 \tag{32}$$

In the classic method, a mean square value including network weights and thresholds is added, and correcting the error performance function with this item can improve the network's generalization ability, as shown in Equation (33):

$$msereg = \gamma \cdot mse + (1 - \gamma)msw \tag{33}$$

where γ is the error performance adjustment rate, and msw can be calculated by Equation (34):

$$msw = \frac{1}{n} \sum_{j=1}^n x_j^2 \tag{34}$$

Equations (32)–(34) can be used to modify the error performance function so that the network obtains a smaller weight and threshold, which forces the network to accordingly become smooth and reduces the phenomenon of overfitting.

2. Early Termination

The early termination method is also called the early stopping method. The basic idea is to stop training when the model occurs overfitting. The method needs a checking data set to recognize the overfitting. After each epoch, the checking set will be brought to the current model. If the checking set's error is increasing continuously, the training would be terminated. The method is shown below.

Dividing the data P_{input} into three subsets yields:

$$P_{input} = \left(P_{input}^s, P_{input}^c, P_{input}^t \right) \tag{35}$$

In Figure 13 and Equation (35), the first subset is the training sample set P_{input}^s , which is used to calculate the weights and thresholds of the gradient and correction network. The second subset is the confirmation sample set P_{input}^c . During training, the error of the confirmation sample set is monitored by this subset. The confirmation sample set's error is typically reduced in the initial stage of training, indicating the catenary sample set's error. When overfitting exists in the network, the error of the confirmation sample set increases significantly. When the number of consecutive increases in the error reaches the specified number of iterations, training is stopped, and the network returns the weight and threshold with the smallest confirmation sample set error. The third subset is the test sample set P_{input}^t . The test sample set error is not used during training, which is used to compare various models.

3.5. Sound Ray Trace Simulation

3.5.1. Basic theory of Sound Ray Trace

The sound speed profile is used in many areas. In real applications, the sound speed profile is used in single beam sounding as an essential correction value. Moreover, the

single beam sounding could be simplified to be a sound ray tracing problem. Therefore, this study will use sound ray simulation to compare the model’s performance between the model’s sound speed profile and measured sound speed profile in practical application.

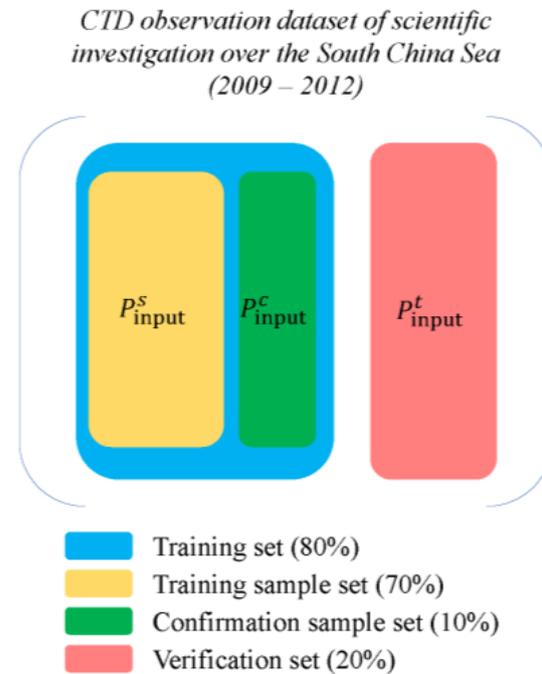


Figure 13. Division of the measured data.

In water, we often use ray acoustics to address the problem of sound propagation. Although ray acoustics are not an exact solution of the wave equation, they can be used as an approximate solution to describe the sound speed profile’s accuracy.

First, the following basic assumptions should be made about the underwater environment.

1. The direction of the sound ray is the direction of sound propagation, and the sound ray is always perpendicular to the wavefront.
2. Sound rays carry energy. At a certain point in the sound field, the sound energy is the superposition of the energy carried by all sound rays arriving at that point.
3. The energy in a group of sound rays is conserved, and there is no transverse energy exchange outside the bundle.

The method used in this experiment is the acoustic line method of the layered medium. Considering the sound speed distribution $c(z)$ of the sound source location and the sea area where the sound source is located, there is a corresponding sound ray for each initial shooting angle α_0 . By continually changing α_0 , a cluster of sound rays can be obtained, and the tracks of these sound rays constitute the acoustic line diagram. In underwater acoustic detection, acoustic line tracing is typically performed according to the sound speed profile to accurately determine underwater points’ coordinates. Therefore, the simulation effect of the BP-Optimized model’s sound speed profile can be verified effectively via acoustic tracing calculations.

3.5.2. Calculation of Sound Ray Trace Parameters

Setting the initial grazing angle α_0 of the sound ray and sound speed c_0 at the sound source, the total sound ray parameters of the layer can be described. The grazing angle α_i of the sound ray at depth z_i is based on Equation (36):

$$\cos \alpha_i = c_i \frac{\cos \alpha_0}{c_0} \tag{36}$$

The horizontal distance of the sound ray in layer i is x_i :

$$t_i = \frac{1}{2\alpha_i c_{i-1}} \left(\ln \frac{1 + \sin \alpha_{i-1}}{1 - \sin \alpha_{i-1}} - \frac{1 + \sin \alpha_i}{1 - \sin \alpha_i} \right) \tag{37}$$

The propagation time of the sound ray in the i layer is t_i :

$$t_i = \frac{1}{2\alpha_i c_{i-1}} \left(\ln \frac{1 + \sin \alpha_{i-1}}{1 - \sin \alpha_{i-1}} - \frac{1 + \sin \alpha_i}{1 - \sin \alpha_i} \right) \tag{38}$$

The total horizontal distance x of the sound ray can be calculated by Equation (37):

$$x = \sum_{i=1}^N x_i \tag{39}$$

The total propagation time t of the sound ray is based on Equation (38):

$$t = \sum_{i=1}^n t_i \tag{40}$$

Equations (36)–(38) can be used to calculate the water’s sound ray’s propagation trajectory.

4. Results and Discussion

This study proposes an optimized LM-BP neural network model to predict the sound field inversion in the South China Sea. Compared to the traditional BP neural network model, the optimized model achieves high training speeds, high training accuracies, low parameter complexity, and good optimization results. In addition, this study verifies the model’s feasibility based on measured sound speed data in the South China Sea. Furthermore, this study used the sound ray tracing method to simulate the underwater sounding devices’ work with BP-Optimized models’ output. Through simulation, it confirms that the model has a high value in practical applications. In conclusion, this study uses a BP-Optimized neural network model to mitigate the problems of slow convergence training, complex parameter settings, low fitting accuracy, and weak generalizability with a traditional BP neural network.

First, we discuss the performance of the proposed sound speed prediction model, which is the optimized BP neural network in this paper. Table 3 shows the results of the proposed model compared to those of other neural networks. The optimized BP neural network is shown to shorten the training time while ensuring sufficient accuracy. AVG refers to the average sound speed profile. Compared with this paper’s model, the root means square error (RMSE) is too large and unsuitable in practical applications. Compared to traditional BP networks, the model achieves marked improvements in precision and computation time. The proposed model achieves approximately 50% higher accuracy than those of the BP-Traditional model (1.7902 m/s to 0.95732 m/s). Computation time also decreases from 612.43 s to 4.231 s. In addition, compared to the genetic algorithm-radial basis function (GA-RBF) model [21], the BP-optimized model achieves a similar accuracy; however, the computation time decreases by approximately 98%. In the GA-RBF model, Hu Jun et al. used a genetic algorithm to confirm the RBF network’s parameters and trained the RBF network with data. Compared with the BP-Optimized model, the GA-RBF model is slower. The reasons are shown below. First, it needs some time to confirm the network’s parameters, and the genetic algorithm’s complexity is not low as we expect. Secondly, the RBF network uses the classic training function, which is significantly slower than the BP-Optimized model. In terms of practical application, for the GA-RBF, it is hard to update the model in real-time.

Table 3. Precision and computation times of various models.

Method	RMSE (m/s)	Time (s)
AVG	2.33475	-
BP-Traditional	1.7903	612.43
GA-RBF	0.96795	272.753
BP-Optimized	0.95732	4.231

The optimized model’s prediction results are shown in Figure 14. The figures show the measured sound speed profile and the predicted sound speed profile, including the sound speed data from 0 m to 1550 m. Each figure’s bottom-right portion shows the set’s position. Due to space limitations, Figure 14 shows only one result out of a total of 12 results, and the remaining results are shown in Figure A1.

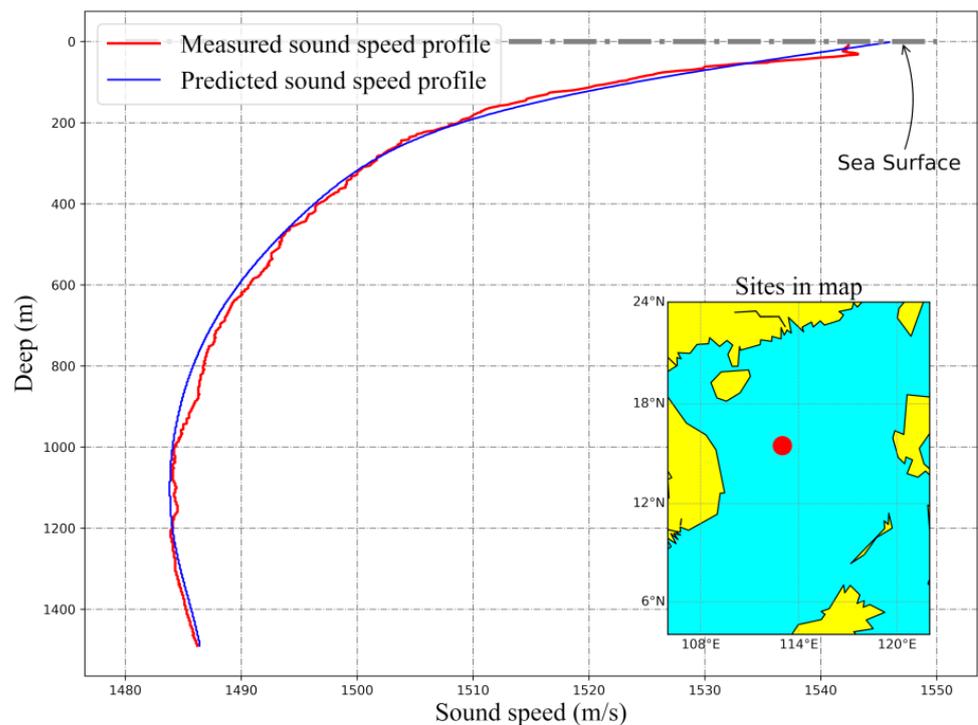


Figure 14. Optimized BP network predicted results at one site.

In Figures 14 and A1, the experimental results show that the optimized BP neural network algorithm achieves good results with South China Sea sound field modeling. The most significant site error is 1.382, the minimum error is 0.671, and the average error is 0.957. The algorithm exhibits poor-fitting accuracy with the surface sound speed; however, more accurate results are produced with the sound speed below the middle layer. Figure 15 shows that the neural network with the LM algorithm and early termination converges quickly. The time required to reach the target accuracy is typically within 6 s, and the number of training epochs does not exceed 300.

Second, the pros and cons of the model are discussed according to different optimization items.

The Levenberg–Marquardt (LM) algorithm is selected during training and is used to replace the calculation of the Hessian matrix in the Gauss–Newton method with an easy-to-calculate Jacobian matrix to improve calculation efficiency. Compared to the Gauss–Newton method and the gradient descent method, the LM algorithm adds μ to the Gauss–Newton method. When μ is large, the algorithm is equivalent to the gradient descent method. When μ is small, the algorithm is equivalent to the Gauss–Newton method.

A good combination of the advantages of the two algorithms allows the model to converge quickly. However, the LM algorithm occupies much more memory than the other methods, and its convergence is too fast, which can lead to early training termination when the early termination method is used.

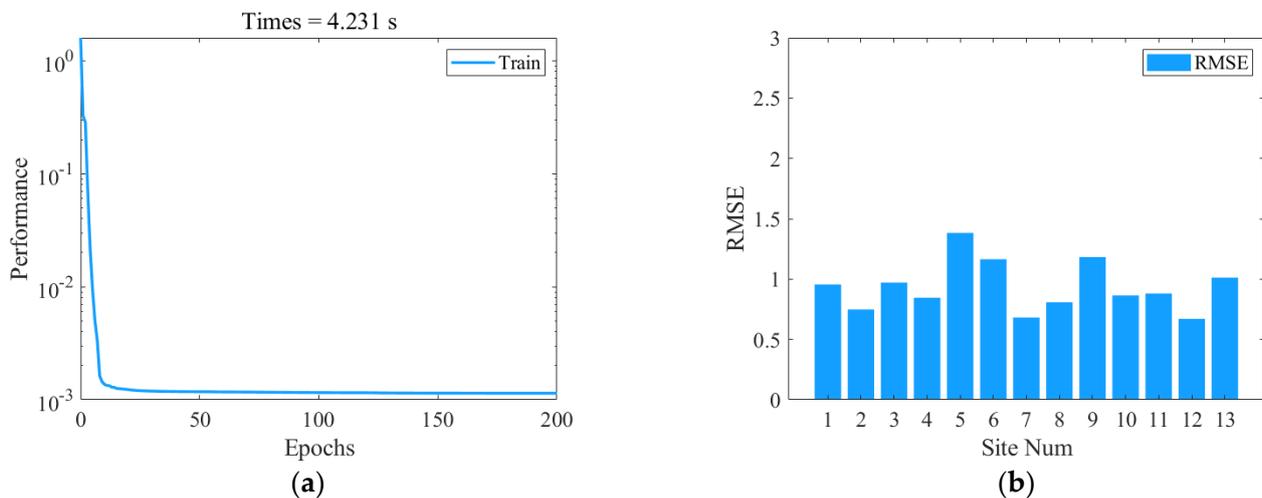


Figure 15. Performance statistic of optimized BP network: (a) predicted RMSE at 13 sites; and (b) performance curve of the optimized BP network.

The momentum term is also used during algorithm learning to (1) adjust the learning step length to mitigate the instability caused by rapid convergence of the LM algorithm, and (2) optimize the problem of artificially reducing the LM algorithm's operation speed in the early termination method. Conversely, increasing the momentum term can also prevent learning from falling into a local minimum and, thus, continuous adjustment of the learning parameter η during training. Although increasing the momentum term balances the training speed, it also increases the time required for a single iteration.

The introduction of the early termination method can effectively prevent overfitting during training. Training is stopped when the number of consecutive increases in the sample set error reaches a specified number of iterations to avoid overtraining. However, there are two problems with the early termination method. First, because the LM algorithm converges too quickly, instability typically occurs, and the error threshold is frequently reached. Stability can be increased by artificially reducing the convergence speed of the LM algorithm and introducing momentum. In addition, the sample set must be sufficiently representative, which requires screening and classifying of the samples. Typical classification methods can use support vector machines and other methods to classify the data and select the representative data after classification.

Thus, due to the merits and demerits of the LM algorithm, momentum term, early termination method, etc., the introduction of a single optimization method in a BP artificial neural network does not typically achieve better optimization results. In this study, the combination of various methods effectively mitigates each method's disadvantages and improves overall model accuracy. However, the introduction of various optimization methods increases model complexity and the complexity of establishing models in different regions.

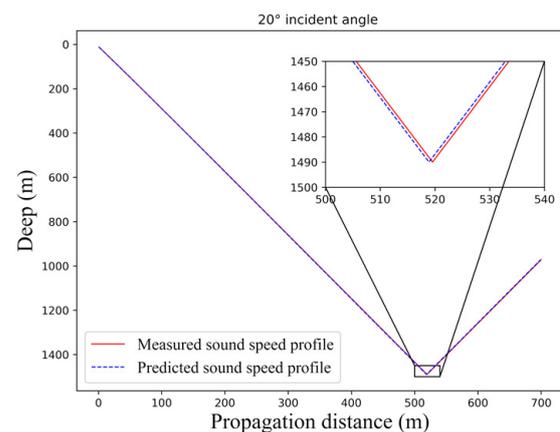
Next, we discuss model performance based on the data verification results.

The model fit to the sound speed data is poor at the surface layer at the verification sites. This result likely occurs due to the following issues. First, ocean sound speed changes most significantly due to changes in temperature. Factors such as exposure to sunlight and seasonal changes have caused complex changes in the ocean surface's sound speed. Second, the sea surface is stirred by wind and waves, resulting in unstable physical and chemical properties in the water at the sea surface. Thus, the surface sound speed is affected by many factors that likely lead to poor sound speed predictions.

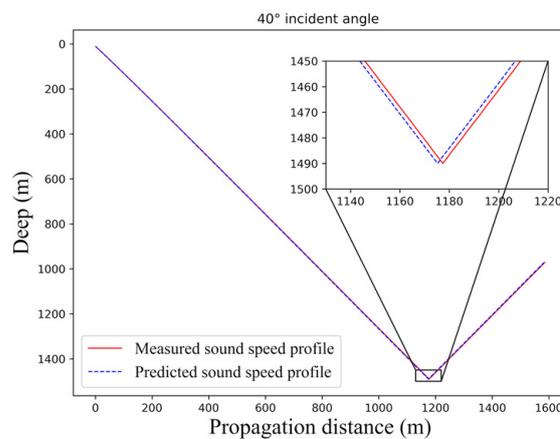
Considering the multiple stations used in this study, some verification stations at the edge of the station show significant errors due to route restrictions. The edge error indicates that the dataset used in this study does not achieve sufficient spatial density, which adversely affects training accuracy.

Finally, the practical applications' simulation of the optimized BP model's sound speed profile is described using the sound ray tracing method in this study. In practical application, the sound speed profile is usually used in marine mapping. Moreover, most measuring devices are based on ray acoustics. The sound ray trace method can effectively simulate the sounding devices working in water, and we use it to evaluate our model. In this section, we used the sound ray trace method to value the paper's optimal model's performance.

Compared to the measured sound speed gradient, when the transducer's opening angle is marginal ($0^\circ \sim 40^\circ$), the model's sound speed profile's horizontal error is below 1 m, which meets the requirements for depth sounding accuracy. In the IHO Standards for Hydrographic surveys (4th Edition), the standard stipulated that over 150 m deep water, the error is limited to 5% depth. However, at a relatively large incident angle, the horizontal error is approximately 5 ~ 10 m, and the error exceeds the limit without correction. Figure 16 shows the accurately measured sound speed's trajectory and the model fitting sound speed's trajectory under two different incident angles.



(a)



(b)

Figure 16. Horizontal errors of the sound ray trace at different incident angles: (a) 20° incident angle, (b) 40° incident angle.

Based on Figure 16, the horizontal sound ray tracing error increases rapidly with the increasing of the incident angle. When the incident angle is 20° , the measured data and model data horizontal sound ray fitting error is below 0.5 m. When the incident angle is 40° , the measured data and model data horizontal sound ray fitting error is below 1 m. In the practical applications of the model of sound speed profile, it can be found that it meets the application requirements given in the IHO.

Thus, this study verifies that the proposed BP-Optimized model can quickly model and fit the regional sound field while meeting accuracy requirements. Compared to the traditional sound speed prediction model, the proposed BP-Optimized model exhibits fast training times, higher accuracies, and strong generalizability. In practical applications, newly acquired data can be incorporated into the prediction model in real time. However, optimization is required based on the following issues. First, from the perspective of the model itself, the data can be pre-classified by various methods such as support vector machines (SVMs) [33]; prediction models will be developed based on different sound field characteristics, and then the quality evaluation methods of training data will be solved by algorithms. This process selects representative data to participate in model construction. Second, when selecting data, the data used in this study do not have good spatial uniformity due to the limitation of the data set. Using uniformly distributed data points will help establish an accurate sound speed prediction model during model training.

5. Conclusions

This study investigated the feasibility of a neural network to calculate a sound speed profile inversion. This study proposed a fast inversion model to calculate the South China Sea sound field based on an optimized BP neural network. The LM algorithm was used to train the algorithm using a neural network, and the momentum term and early termination method are also introduced. Then, the paper used the sound ray trace method to simulate the real application. When we used the sound ray trace method to simulate the acoustic devices, the simulation showed that the paper's sound speed prediction model has a high precision and application value. The proposed model calculated an accurate sound speed inversion prediction, and a comparison of the proposed model and other models is shown below:

1. The proposed model only requires the time, the latitude and longitude, and the depth of the layer measurement to fit any sound speed profile within the model range within a given duration. The accuracy of this model can meet the accuracy requirements of most practical applications.
2. Compared to a traditional BP neural network, the improved neural network introduces an early termination mechanism that effectively mitigates overfitting. In the traditional method, the neural network can approximate a linear function with arbitrary precision. When the target accuracy is high, the training model will often cause overfitting with the training data and not fit non-training data. The early termination mechanism can effectively determine the optimal training accuracy.
3. The improved neural network algorithm uses the LM algorithm, which has the same generalizability as the gradient descent method and the Gauss–Newton method's local convergence. The improved algorithm also achieves fast convergences and high training accuracies. Compared to the traditional neural network, the improved algorithm significantly reduces training time.
4. Table 3 shows the results of the proposed model compared to those of other neural networks. The optimized BP neural network markedly reduces training time while ensuring sufficient accuracy. Compared to traditional BP networks, the proposed model markedly improves precision and reduces computation time. Compared to the RBF neural network, the proposed model achieves similar accuracies but decreases computation time by approximately 98%.

In conclusion, this paper introduced an optimized LM-BP artificial neural network model to achieve the rapid modeling of the sound speed profile. Compared with other studies, it has more significant value in practical application. First, we presented a sound speed prediction model, which could satisfy the accuracy requirement and sharply reduce training time. Secondly, we used the sound ray tracing method to verify the model’s practical application value. The simulation result shows that the prediction model is sufficiently precise in engineering applications.

Author Contributions: Conceptualization, J.H.; Data curation, J.S.; Formal analysis, X.M.; Funding acquisition, Y.L.; Investigation, Q.-Q.L.; Methodology, J.H. and Y.-Y.L.; Software, J.S.; Supervision, Y.L.; Visualization, X.M., Q.-Q.L. and Y.-Y.L.; Writing—Original draft, J.H.; Writing—Review and editing, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National College Students Innovation and Entrepreneurship Training Program, grant number S202010424021.

Acknowledgments: Thanks to the South China Sea Institute of Oceanology, the Chinese Academy of Sciences for providing data support for this study and the members of the project for their contributions to this paper. Finally, we express thanks to Professor Luo Yu for his guidance in this research.

Conflicts of Interest: The authors declare no conflict of interest. The sponsors had no role in the design, execution, interpretation, or writing of the study.

Appendix A

Optimized BP network predicted results, at 12 sites, are shown below:

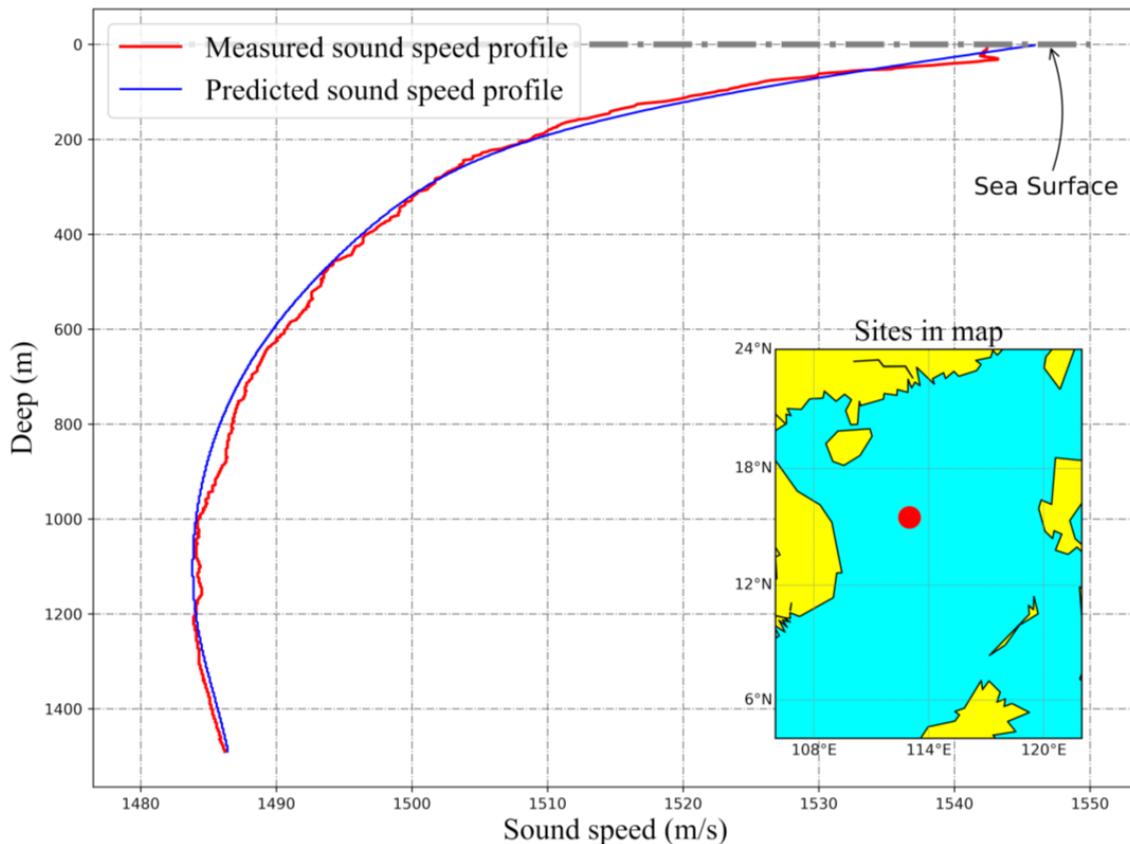


Figure A1. Cont.

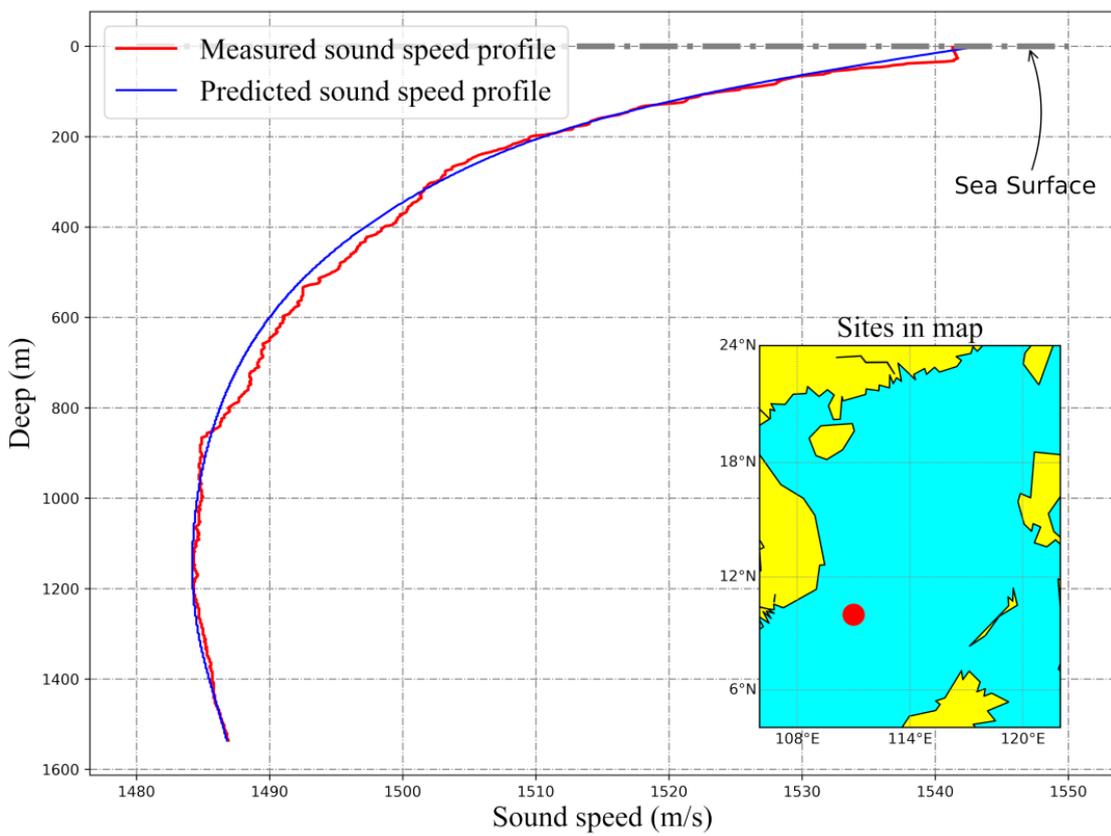
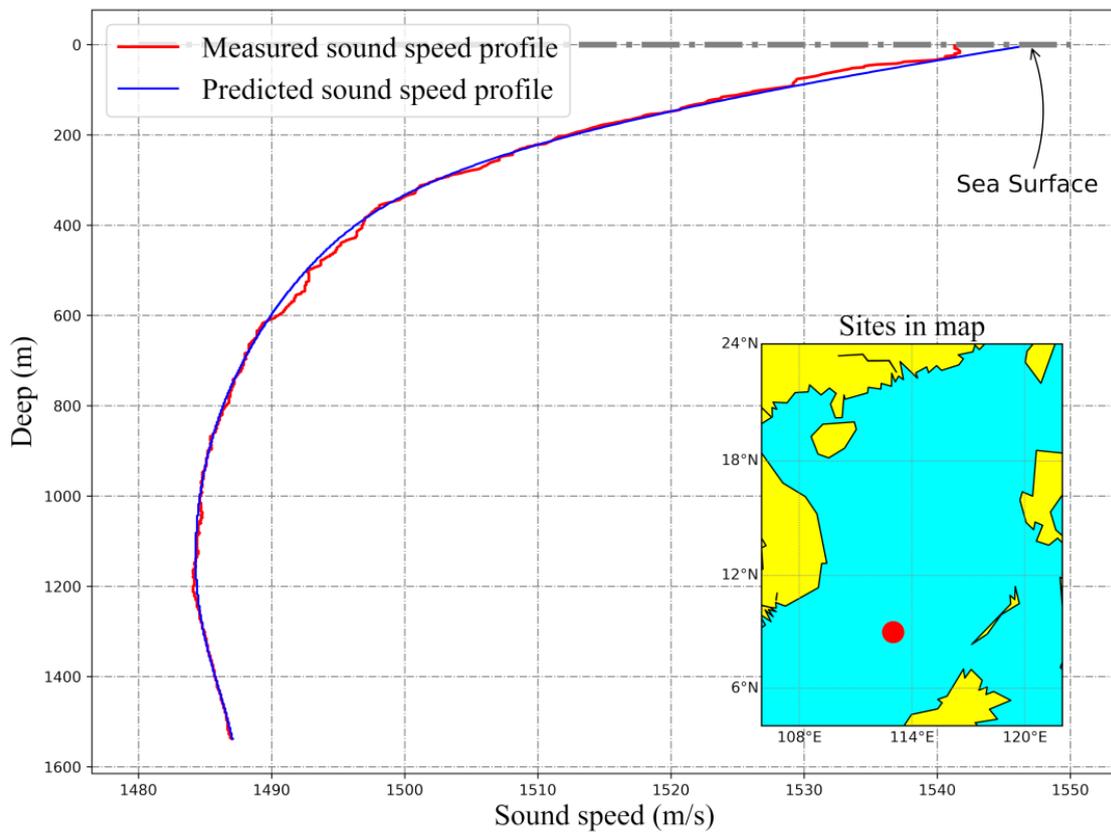


Figure A1. Cont.

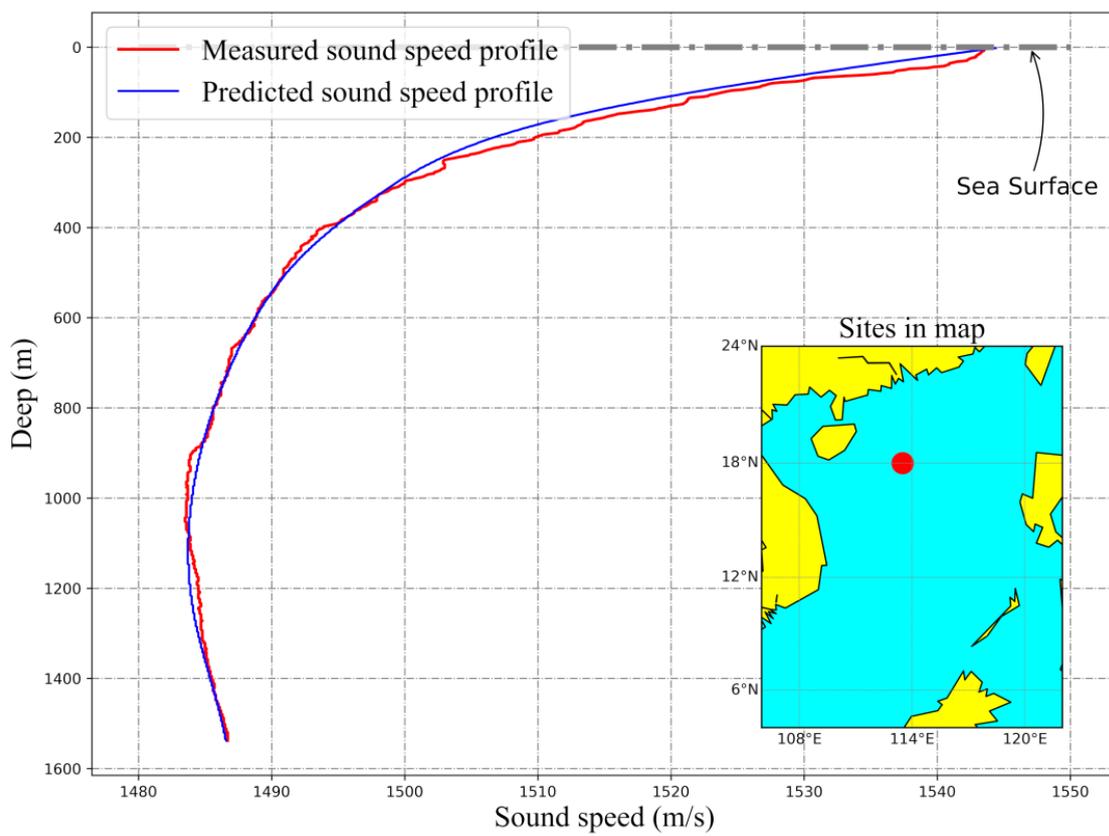
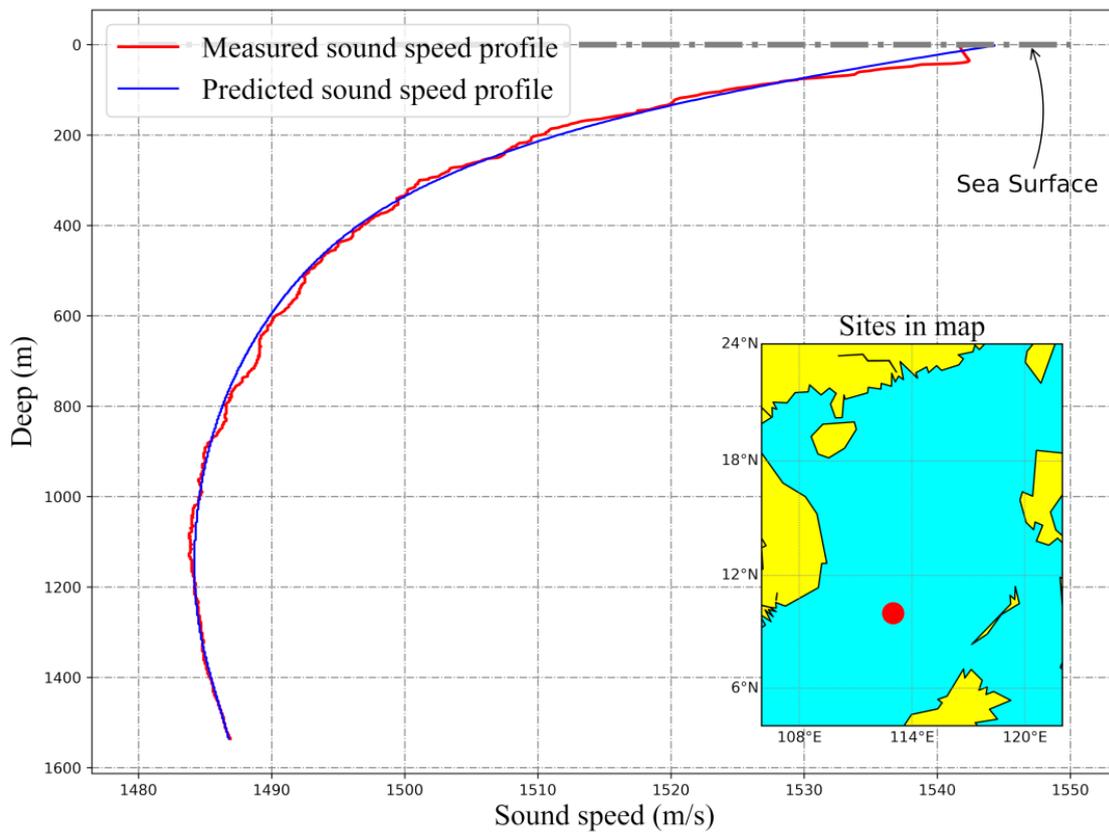


Figure A1. Cont.

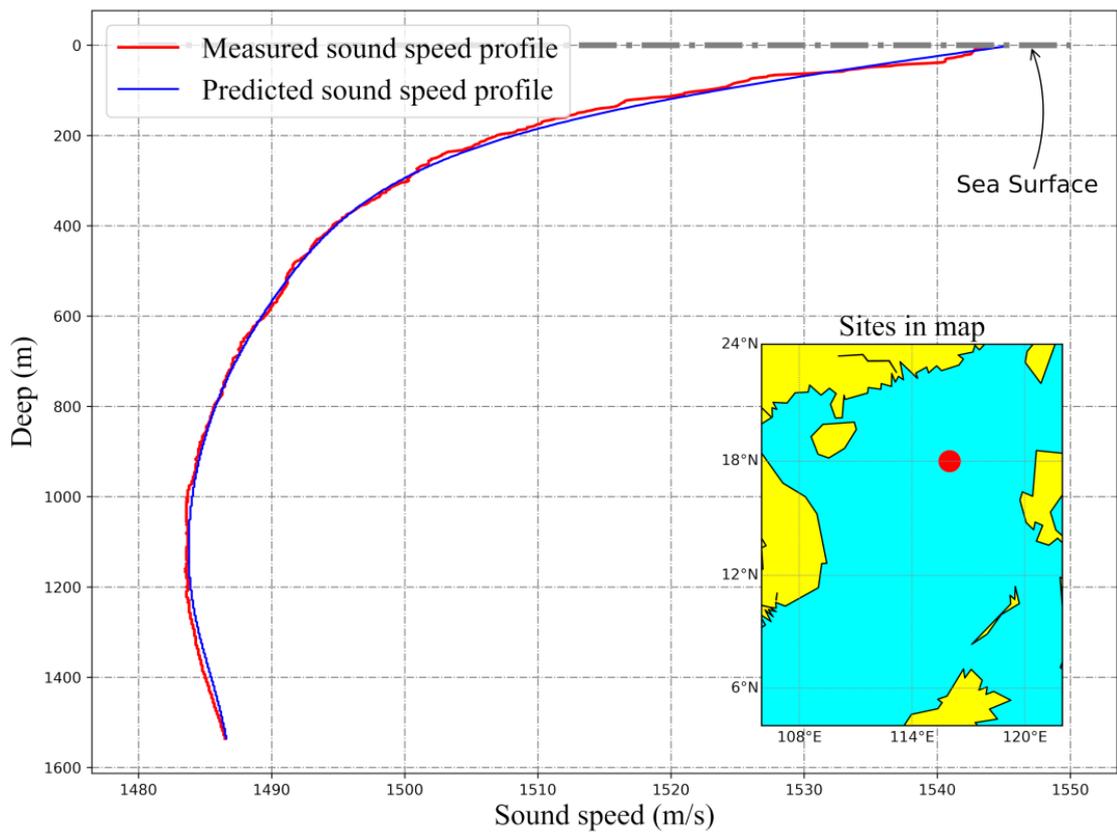
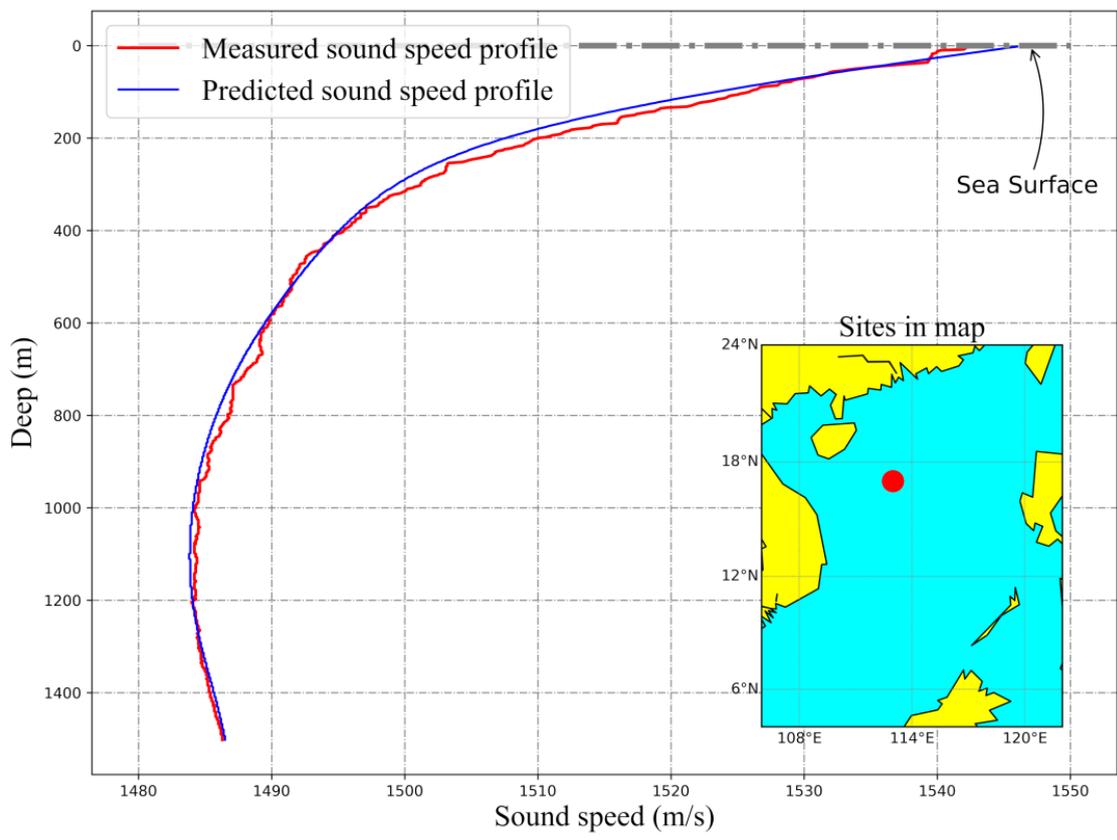


Figure A1. Cont.

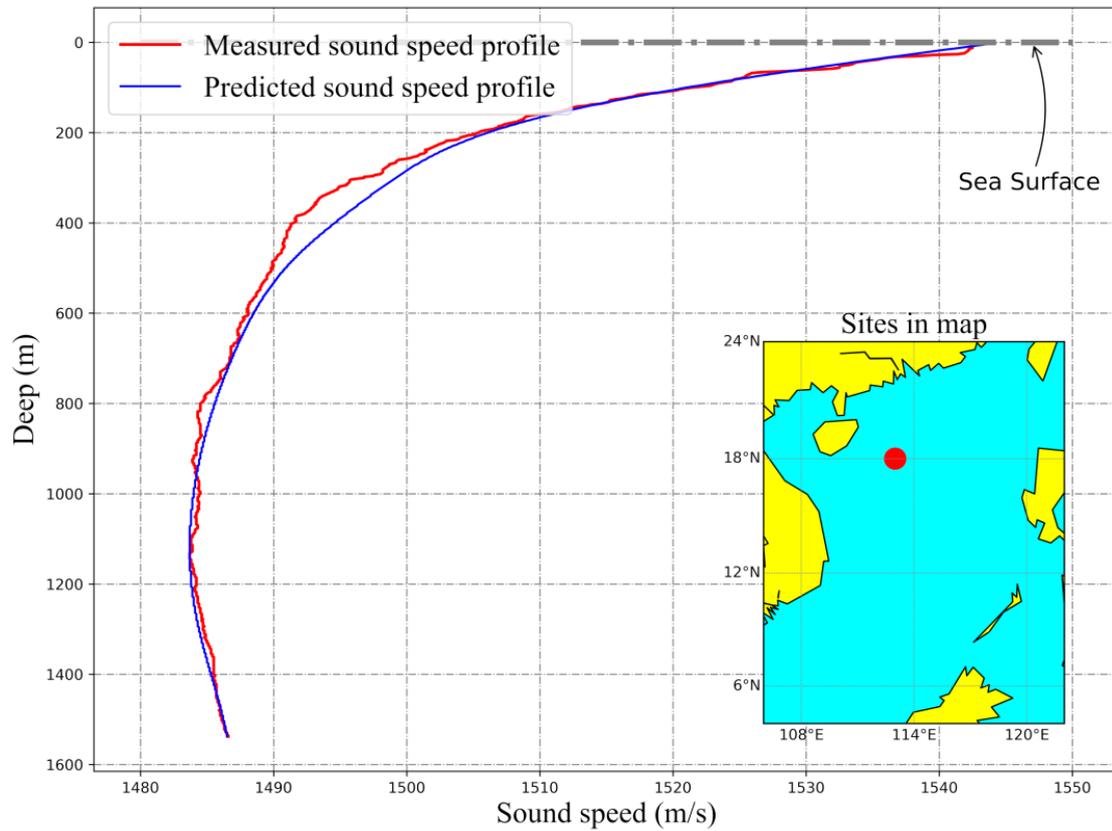
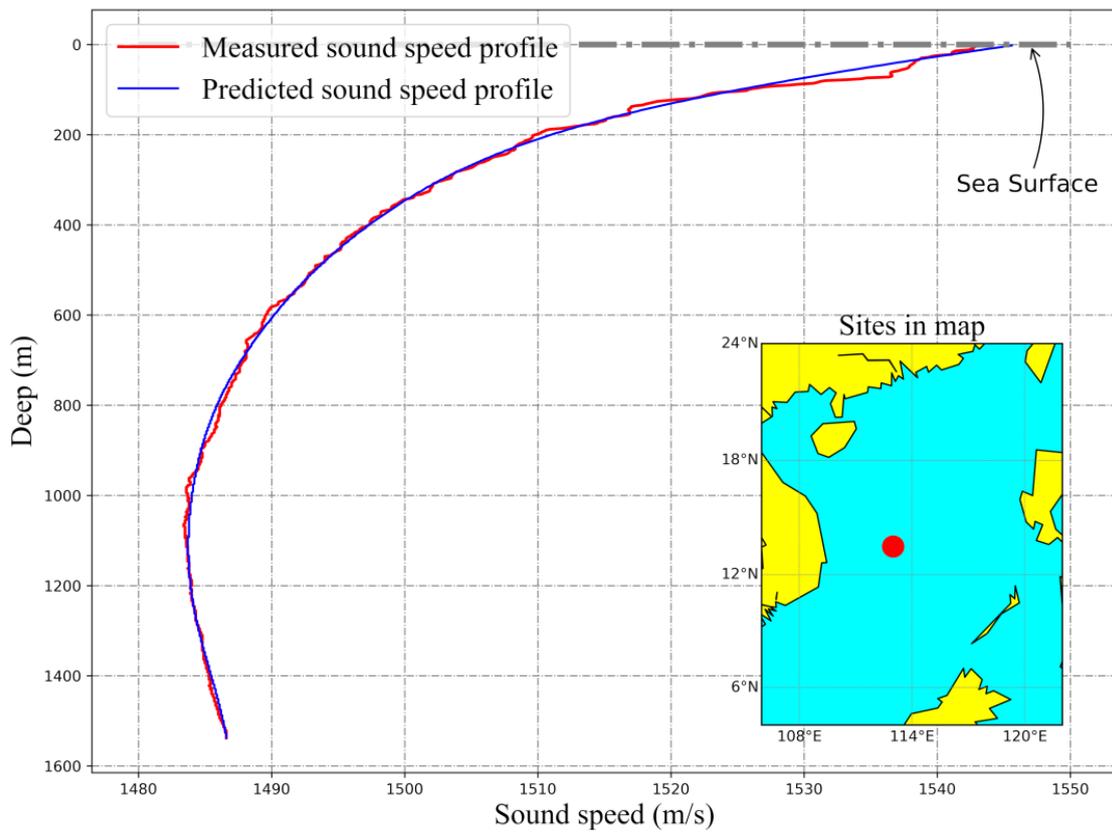


Figure A1. Cont.

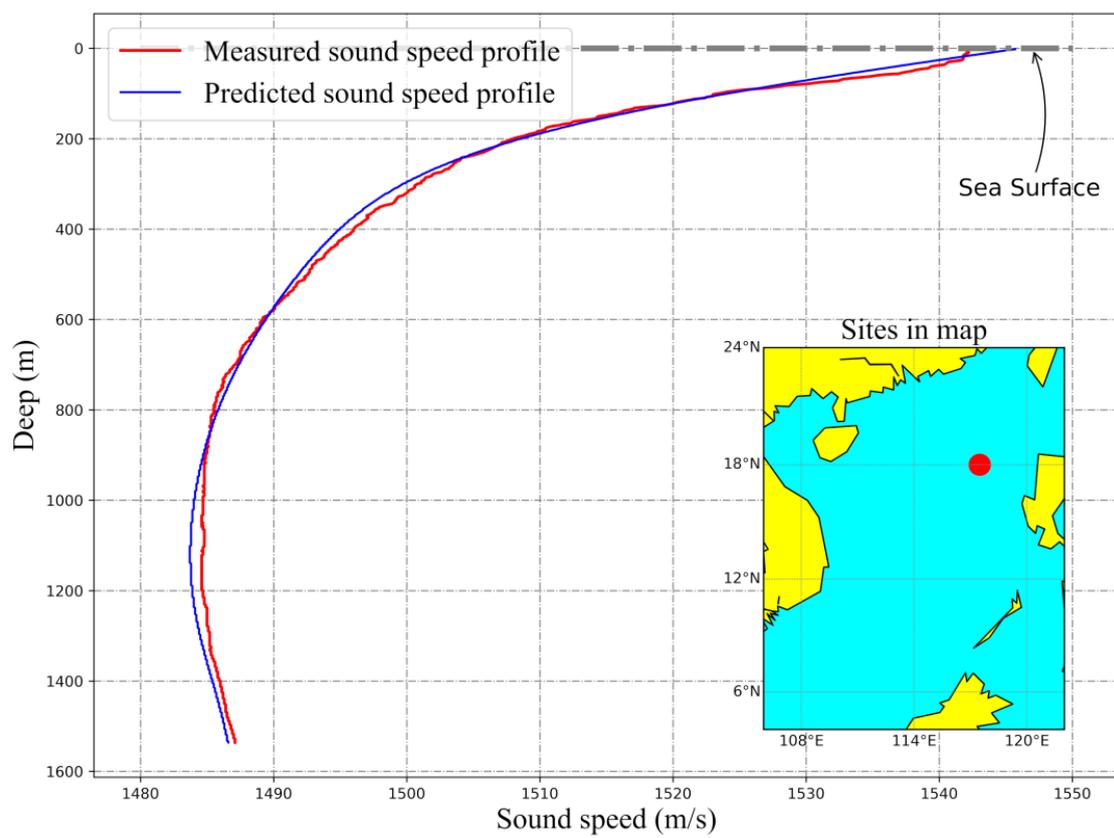
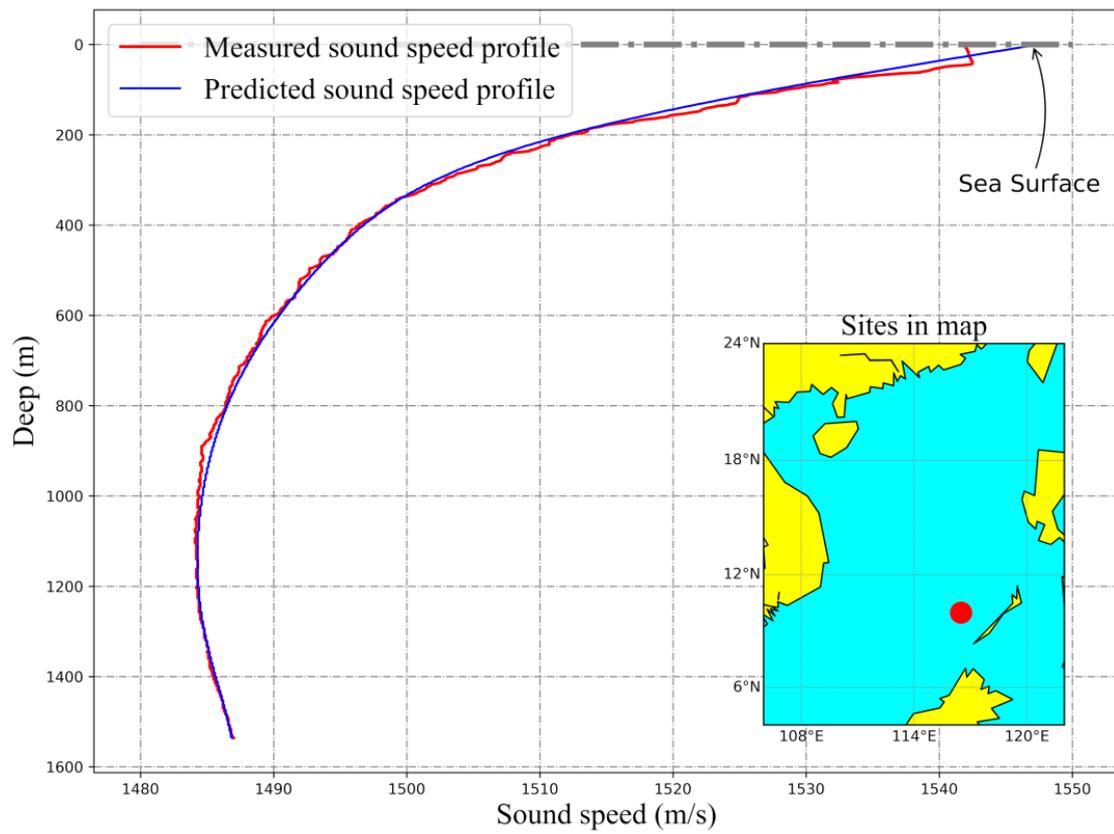


Figure A1. Cont.

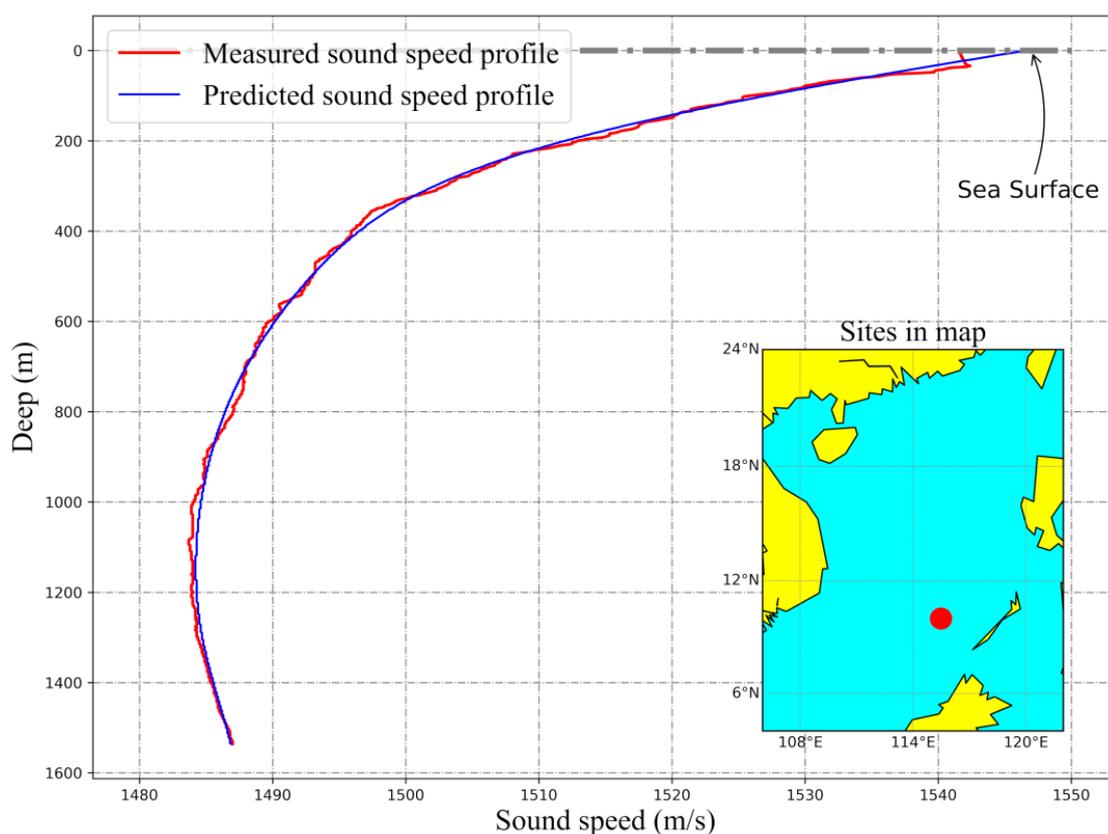


Figure A1. Optimized BP network predicted results at 12 sites.

References

- Li, Y.Y.; Da, L.L.; Song, J.; Hua, H. Research on Underwater Acoustic Environment Simulation and Visualization. *J. Geomat. Sci. Technol.* **2009**, *26*, 8–11.
- Li, Y.; Da, L.; Han, M.; Jin, C.-B. Underwater Acoustic Environment Rapid Assessment Modeling and Simulation. *J. Syst. Simul.* **2012**, *24*, 1156–1160.
- Del Grosso, V.A. New Equation for the speed of sound in natural waters (with comparisons to other equations). *Acoust. Soc. Am.* **1974**, *4*, 1084–1091. [[CrossRef](#)]
- Denisov, D.A.; Abramov, A.V.; Abramova, E.P. A method for calculating the speed of sound in seawater. *Acoust. Phys.* **2003**, *4*, 413–419. [[CrossRef](#)]
- Carnes, M.R.; Mitchell, J.L.; De Witt, P.W. Synthetic temperature profiles derived from Geosat altimetry: Comparison with air-dropped expendable bathythermograph profiles. *J. Geophys. Res. Ocean.* **1990**, *95*, 17979–17992. [[CrossRef](#)]
- Zhou, H. Inversion of Sound Speed Profile in Dynamic Ocean Environment. Master's Thesis, Zhejiang University, Hangzhou, China, 2013.
- Munk, W.; Worcester, P.; Wunsch, C. Ocean acoustic tomography. In *Ocean Acoustic Tomography*; Cambridge University Press: Cambridge, UK, 2009.
- Leblanc, L.R.; Middleton, F.H. An underwater acoustic sound velocity data model. *J. Acoust. Soc. Am.* **1980**, *67*, 2055–2062. [[CrossRef](#)]
- Davis, T.M.; Countryman, K.A.; Carron, M.J. Tailored acoustic products utilizing the NAVOCEANO GDEM (a generalized digital environmental model). In *Proceedings, 36th Naval Symposium on Underwater Acoustics*; Naval Ocean Systems Center San Diego, Naval Ocean Systems Center: San Diego, CA, USA, 1986.
- Wang, Y. The Sound Speed Profile Reconstruction and Its Software Realization Based on composition of Empirical Orthogonal Function. Master's Thesis, Harbin Engineering University, Haerbin, China, 2011.
- Zhang, X.; Zhang, Y.; Zhang, S.; Wu, S. A comparison between the EOF and GDEM methods for reconstruction of sound speed profile. *Appl. Acoust.* **2010**, *29*, 115–121.
- Teague, W.J.; Carron, M.J.; Hogan, P.J. A Comparison between the Generalized Digital Environmental Model and Levitus climatologies. *J. Geophys. Res. Space Phys.* **1990**, *95*, 7167–7183. [[CrossRef](#)]
- Davis, R.E. Predictability of Sea Surface Temperature and Sea Level Pressure Anomalies over the North Pacific Ocean. *J. Phys. Oceanogr.* **1976**, *6*, 249–266. [[CrossRef](#)]
- Park, J.; Kennedy, R. Remote sensing of ocean sound speed profiles by a perceptron neural network. *IEEE J. Ocean. Eng.* **1996**, *21*, 216–224. [[CrossRef](#)]

15. Zhang, X.; Zhang, Y.; Zhang, S.; Wu, S. A comparison between the EOF and GDEM method in reconstruction of sound speed profile. *Tech. Acoust.* **2009**, *28 Pt 2*, 6–9.
16. Rashid, T.L. The title of the cited contribution. In *Make Your Own Neural Network: A Gentle Journey through the Mathematics of Neural Networks and Making Your Own Using the Python Computer Language*; CreateSpace Independent Publ.: London, UK, 2007.
17. Zhang, W.X.; Gao, G.D.; Mu, G.Y. Study on the Model of Salinity Based on Back-Propagation Artificial Neural Network. *Ocean Technol.* **2006**, *25*, 39–41.
18. Hu, H. Application of Back-Propagation Neural Network in Establishing Velocity Field. *Hydrogr. Surveying Charting* **2015**, *35*, 67–70.
19. Ai, R.F.; Cheng, J.; OuYang, J.; Yang, J. On-line retrieval methodology for sound speed profile of sea area. *J. Comput. Appl.* **2015**, *35*, 327–330.
20. Najah, A.; El-Shafie, A.; Karim, O.A. Application of artificial neural networks for water quality prediction. *Neural Comput. Appl.* **2013**, *22*, 187–201. [[CrossRef](#)]
21. Hu, J.; Xiao, Y.; Zhang, D.; Leng, L. Inversion prediction method for sound profile. *Adv. Mar. Sci.* **2019**, *37*, 245–254.
22. Xu, C.; Li, S.; Chen, R.; He, Y.; Chen, S. CTD observation dataset of scientific investigation over the South China Sea (2009–2012) [J/OL]. *China Sci. Data* **2016**, *1*. [[CrossRef](#)]
23. Jensen, F.B. Optimum frequency of propagation in shallow water environments. *J. Acoust. Soc. Am.* **1983**, *73*, 813. [[CrossRef](#)]
24. Tindle, C.T.; Guthrie, K.M. Rays as interfering modes in underwater acoustics. *J. Sound Vib.* **1974**, *34*, 291–295. [[CrossRef](#)]
25. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
26. Rubanov, N.S. The layer-wise method and the backpropagation hybrid approach to learning a feedforward neural network. *IEEE Trans. Neural Netw.* **2000**, *11*, 295. [[CrossRef](#)]
27. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
28. Vealanturf, L.P.J. *Analysis and Applications of Artificial Neural Networks*; Prentice-Hall, Inc. Press: Hoboken, NJ, USA, 1995.
29. Zhang, Q.; Li, X. A New Method to Determine Hidden Node Number in Neural Network. *J. Jishou Univ. Nat. Sci. Ed.* **2002**, *23*, 89–91.
30. Stephens, M.A. Kolmogorov–Smirnov-type Tests of Fit. *J. Am. Stat. Assoc.* **2014**, *69*, 730–737. [[CrossRef](#)]
31. Shen, H.; Wang, Z.; Gao, C.; Qin, J.; Yao, F.; Xu, W. Determining the number of BP neural network hidden layer units. *J. Tianjin Univ. Technol.* **2008**, *24*, 13–15.
32. Jadid, M.N.; Fairbairn, D.R. The application of neural network techniques to structural analysis by implementing an adaptive finite-element mesh generation. *AI EDAM* **1994**, *8*, 177–191. [[CrossRef](#)]
33. Li, Y.; Wang, J.; Gao, T.; Sun, Q.; Zhang, L.; Tang, M. Adoption of Machine Learning in Intelligent Terrain Classification of Hyperspectral Remote Sensing Images. *Comput. Intell. Neurosci.* **2020**, *2020*, 1–13. [[CrossRef](#)]