sensors

Article

Time–Frequency Mask-Aware Bidirectional LSTM: A Deep Learning Approach for Underwater Acoustic Signal Separation

Jie Chen *, Chang Liu, Jiawu Xie, Jie An and Nan Huang

National Key Laboratory of Science and Technology on Communication, University of Electronic Science and Technology of China, Chengdu 610000, China; cliu.wcom@uestc.edu.cn (C.L.); 201622260214@std.uestc.edu.cn (J.X.); 202022220232@std.uestc.edu.cn (J.A.); 202121220218@std.uestc.edu.cn (N.H.)

* Correspondence: jiechen@uestc.edu.cn

Abstract: Underwater acoustic signal separation is a key technique for underwater communications. The existing methods are mostly model-based, and cannot accurately characterize the practical underwater acoustic communication environment. They are only suitable for binary signal separation and cannot handle multivariate signal separation. However, recurrent neural networks (RNNs) show a powerful ability to extract the features of temporal sequences. Inspired by this, in this paper, we present a data-driven approach for underwater acoustic signal separation using deep learning technology. We use a bidirectional long short-term memory (Bi-LSTM) approach to explore the features of a time–frequency (T-F) mask, and propose a T-F-mask-aware Bi-LSTM for signal separation. Taking advantage of the sparseness of the T-F image, the designed Bi-LSTM network is able to extract the discriminative features for separation, which further improves the separation performance. In particular, this method breaks through the limitations of the existing methods and not only achieves good results in multivariate separation but also effectively separates signals when they are mixed with 40 dB Gaussian noise signals. The experimental results show that this method can achieve a 97% guarantee ratio (PSR), and the average similarity coefficient of the multivariate signal separation is stable above 0.8 under high noise conditions. It should be noted that our model can only handle known signals such as test signals for calibration.

Keywords: blind source separation; binary mask; deep learning; underwater acoustic signal

check for updates

Citation: Chen, J.; Liu, C.; Xie, J.; An, J.; Huang, N. Time–Frequency Mask-Aware Bidirectional LSTM: A Deep Learning Approach for Underwater Acoustic Signal Separation. *Sensors* **2022**, *22*, 5598. https://doi.org/10.3390/s22155598

Academic Editor: Aimé Lay-Ekuakille

Received: 7 April 2022
Accepted: 19 July 2022
Published: 26 July 2022

1. Introduction

At present, underwater acoustic communication [1] mainly uses sonar technology to detect, locate and identify underwater targets. However, sonar technology has to overcome effects from noise such as ship noise and ocean noise [2–4]. Therefore, a method of reducing the impact of noise is the most critical part of underwater communication. Source separation technology is a good method of reducing noise [5–8] that has attracted a great deal of attention from researchers in both academia and industry. Among these source separation methods, non-negative matrix factorization (NMF) [9] is one method that can be used to separate source signals. This method converts complex and sensitive feature extraction problems into non-negative matrix dimensionality reduction problems by extracting a set of basis vectors describing the underlying features of the target. However, the correlation of its features causes more similar-feature redundancy in the basis matrix and weakens the feature coefficients in the linear representation, which is not conducive to target recognition. Blind source separation (BSS) is also a classical method [10–12], consisting of a mathematical model, an objective function, a separation algorithm and evaluation criteria [13,14]. In research into the BSS algorithm, two approaches are always studied and employed. One is based on independent component analysis (ICA) [15], which works well when the number of sources N is less than or equal to the number of sensors

M. The use of ICA is not limited to linear instantaneous mixing; it is also used to solve the separation problem for convolutional mixing and even nonlinear mixing. The other relies on the sparseness of source signals, which works well when N is greater than M, e.g., the binary T-F mask approach [16]. The binary T-F mask approach extracts a signal by calculating the binary masking matrix of the signal. It has the advantage of real-time operation, and in recent years it has also been applied to underwater acoustic separation in combination with underwater sound characteristics.

In view of the underdetermination in underwater acoustic communication, this study considers the binary time–frequency mask method based on sparsity. The traditional binary T-F mask method chooses features which are performed manually by using the observation signals. Due to the outliers and distribution of anisotropic variance, the traditional feature extraction method has certain limitations. It can only be used in binary signal separation, as the effect is poor in multiple signal separation and it cannot meet the requirements of separation accuracy. At present, the improvement of the binary T-F masking method remains a matter of feature design [17–19]. However, it is not easy for human experts to design good features. These artificial features are easily affected by outlier problems and have strict requirements regarding the selection of source location. As an alternative, in addition to traditional binary T-F masking, the method of extracting the original features of the underwater acoustic source using a deep neural network has shown good performance. This method has been used to solve image recognition, natural language processing (NLP) and even communication problems [20]. The deep learning approach [21–23] also represents a breakthrough in the separation of signals. Therefore, we extract the features of the underwater acoustic signals by means of a deep learning approach. The main contributions of this work are as follows:

(1) We propose a deep learning method based on Bi-LSTM. This method uses the powerful feature extraction capability of RNN and not only improves the performance in separating binary signals but also achieves good results in ternary or multivariate signal separation experiments. This overcomes the limitations of the previous separation of single targets from deep learning sources.

(2) We improved the training sample using the idea of embedding, i.e., embedding each T-F point into a high-dimensional space so that each T-F point can be represented as a vector, and then adding energy-based reference labels to the training sample. This makes the T-F points of different sources more distinct and makes clustering easier in the process of neural network learning.

(3) We carried out many experiments on the separation performance of this method by using unknown randomly generated noise and the marine noise actually collected. The experimental results show that with an increase in the number of clusters K, the effect of this method in separating noise improves further. It was proved that this method has the ability to reduce the noise impact of passive sonar platforms and to improve the recognition rate of underwater targets, which is significant for improving the performance of sonar positioning, detection and identification.

The rest of the paper is organized as follows. In Section 2, we introduce a traditional system model for underwater acoustic source separation. Then, in Section 3, we present a description of the proposed approach, including offline training and online testing. Section 4 presents the experiments. Finally, conclusions are drawn in Section 5.

2. Mainstream Method: Binary Time–Frequency Masking Method

The binary T-F mask approach separates the underwater acoustic signals according to the auditory masking, using the underwater acoustic source that dominates the energy in a certain T-F domain. Although the target signals received by the system have varying degrees of frequency-band overlap, the main energy of different target signals is usually hidden in different frequency bands. Hence, the binary mask approach can use this property to realize underwater acoustic signal separation by clustering the T-F bins. To cluster the

T-F bins, the traditional method uses the observation signals and calculates manually to obtain the features.

2.1. Restrictions on Using Existing Methods

The use of binary T-F masking techniques must satisfy the sparsity condition. Since the sound signal is generally not sparse in the time domain, it must be transformed into the T-F domain by some transformation [24,25]. However, in the actual separation process of underwater acoustic signals, it is found that the energy of different underwater acoustic radiation signals is usually concentrated in different frequency bands, and the target radiation signals received by the system will show different frequency-band aliasing phenomena. The study found that as long as the underwater acoustic signal can satisfy the absolute dominant condition for the energy, the binary T-F masking algorithm can be used to achieve separation. This condition is written as:

$$|X_i(t,f)| \gg |X_j(t,f)|, i \neq j, \forall t, f, \tag{1}$$

where $X_i(t,f)$ is the short-time Fourier transform (STFT) of signal $x_i(t)$. Using STFT, signals in the time domain can be transformed to the T-F domain, which can satisfy the property of sparsity. Geometric features for clustering are calculated based on this constraint.

This condition can also be understood as representing the fact that the overlap of the T-F domain is a relatively small portion of one of the underwater acoustic signals, so that ignoring the information in this part does not affect the recovery of the entire signal.

2.2. Signal Separation Steps in Underdetermined Case

This approach is summarized in Figure 1. Based on the sparsity condition of absolute dominance of the energy, in the underdetermined case, the idea of using the binary T-F masking method for water acoustic blind separation is as follows:

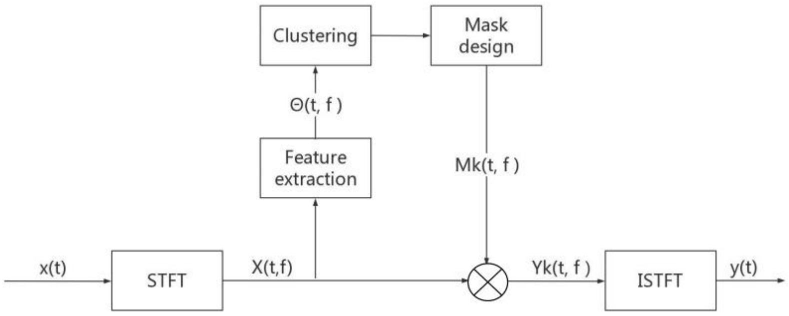


Figure 1. Example of T-F mask approach.

(1) STFT. Let the sampling frequency of the observation signal be f_s and convert the time domain signal $x(t)$ into the T-F domain representation by using the T-point STFT transform:

$$X(t,f) = \sum_{r=-T/2}^{T/2-1} x(r+tL)win(r)e^{-j2\pi fr}, \tag{2}$$

where t is the time in seconds, f is the frequency in Hertz, T is the length of the window and L is the moving length of the window. Here, $win(r)$ represents the window function. Commonly used functions are the rectangular window, Hanning window and Hamming window. In the subsequent inverse short-time Fourier transform (ISTFT), we used the Hanning window for the transformation, to ensure consistent parameters.

(2) Feature extraction. The source signal $X(t,f)$ satisfying the sparse condition is obtained using the STFT transform, and the feature vector $\Theta(t,f)$ is calculated therefrom. In this eigenvector, there are differences between different sources which can be measured by distance. The eigenvector $\Theta(t,f)$ is generally composed of the geometric characteristic

magnitude $\alpha(t,f)$ and the phase difference $\phi(t,f)$ between the observed signals. Taking two observation signals $X_1(t,f)$ $X_2(t,f)$ as an example, the eigenvector $\Theta(t,f)$, the order of magnitude $\alpha(t,f)$ and the phase difference $\phi(t,f)$ can be calculated using the following equations:

$$\Theta(t,f) = [\alpha(t,f), \phi(t,f)], \tag{3}$$

$$\alpha(t,f) = \frac{|X_2(t,f)|}{|X_1(t,f)|}, \tag{4}$$

$$\phi(t,f) = \arg\left(\frac{X_2(t,f)}{X_1(t,f)}\right). \tag{5}$$

The phase difference is usually normalized to avoid frequency sequencing problems, and the above equation can be written as:

$$\phi(t,f) = \frac{1}{2\pi f}\arg\left(\frac{X_2(t,f)}{X_1(t,f)}\right). \tag{6}$$

Expanded to the case where there are multiple observation signals, the order of magnitude $\alpha(t,f)$ and the phase difference $\phi(t,f)$ are expressed as:

$$\alpha(t,f) = \left[\frac{|X_1(t,f)|}{A(t,f)}, \ldots \frac{|X_n(t,f)|}{A(t,f)}\right], \tag{7}$$

$$A(t,f) = \sqrt{\sum_{j=1}^{n} |X_j(t,f)|^2}, \tag{8}$$

$$\phi(t,f) = \left[\frac{1}{\beta_1 f}\arg\left(\frac{X_1(t,f)}{X_B(t,f)}\right), \ldots, \frac{1}{\beta_n f}\arg\left(\frac{X_n(t,f)}{X_B(t,f)}\right)\right], \tag{9}$$

where, $A(t,f)$ is the normalization coefficient of the order of magnitude; $\beta_j = \beta = 4\pi d_{max}/c, j = 1, \ldots, n$ is the weight coefficient of the phase difference, subscript B represents the label of the reference observation signal, c represents the sound propagation speed and d_{max} represents the maximum distance between the reference observation signal and other observation signals.

We express $\Theta(t,f)$ as a plural form with the following equation:

$$\tilde{\Theta}_i(t,f) = |X_i(t,f)| \exp\left[j\frac{\arg(X_i(t,f)/X_B(t,f))}{\beta_i f}\right]. \tag{10}$$

Normalization of the above equation yields a eigenvector representation of the observed multiple signal:

$$\Theta_i(t,f) = \tilde{\Theta}_i(t,f)/\|\tilde{\Theta}_i(t,f)\|, \tag{11}$$

$$\mathbf{\Theta}(t,f) = [\Theta_1(t,f), \ldots \Theta_n(t,f)]^{\mathrm{T}}. \tag{12}$$

From Equations (10)–(12), we know that $\mathbf{\Theta}(t,f)$ is influenced by $|X_i(t,f)|$, $X_B(t,f)$, β_i and f.

(3) Cluster analysis. Clustering the feature vector $\mathbf{\Theta}(t,f)$ can result in m clusters $C_1, \ldots, C_m$ corresponding to m source signals. Past clustering methods include manual clustering [16], kernel density estimation [26] and the maximum likelihood (ML)-based gradient search method [27]. Because K-means clustering has the characteristic of simple, convenient and fast convergence, it has become the most commonly used method for cluster analysis. K-means can minimize the sum Y of the Euclidean distances (EDs) of each

source signal and the corresponding cluster center c_k, and can automatically divide the samples into m clusters. The equation is expressed as:

$$\Upsilon = \sum_{k=1}^{m} \Upsilon_k, \tag{13}$$

$$\Upsilon_k = \sum_{\mathbf{\Theta}(t,f)\in C_k} \|\mathbf{\Theta}(t,f) - c_k\|^2. \tag{14}$$

First, m cluster centers $c_1, c_2, \ldots, c_m$ are randomly initialized, and each feature vector is assigned by iterating Equation (15). Then, the feature vector $\Theta(t,f)$ closest to the mean vector c_k is found and assigned as a cluster:

$$C_k = \{\Theta(t,f) | k = \underset{k}{\mathrm{argmin}} \|\Theta(t,f) - c_k\|^2\}, \tag{15}$$

Then, we calculate the mean of all feature vectors belonging to c_k and correct the cluster center:

$$c_k \leftarrow E[\mathbf{\Theta}(t,f)]_{\mathbf{\Theta}\in C_k}, \tag{16}$$

Substituting the updated mean vector into Equations (13) and (14) enables calculation of the objective function Υ. If Υ converges, then the set $C_k, k = 1,2,\ldots,m$ corresponding to each source is obtained after the iteration ends.

(4) Binary T-F masking. Using the results obtained by clustering, a binary T-F masking matrix is constructed. The binary T-F masking matrix is a matrix consisting of values 0 and 1, whose size is consistent with the T-F matrix. This is similar to the binary test in spectrum sensing [28–31]. The matrix sets the mask value to 1 or 0 according to whether each T-F point belongs to the target signal, indicating whether the T-F-point information belongs to the source signal.

$$M_k(t,f) = \begin{cases} 1, & \mathbf{\Theta}(\mathbf{t},\mathbf{f}) \in C_k \\ 0, & \text{others} \end{cases},$$

Substituting into the following equation gives the spectrum of the estimated signal:

$$Y_k(t,f) = M_k(t,f)X(t,f). \tag{17}$$

(5) Inverse short-time Fourier transform (ISTFT). After obtaining the T-F domain estimation, the final step must complete the recovery of the time domain signal $y_k(t)$ using ISTFT and the overlap retention method [32]:

$$y_k(t) = \frac{1}{A}\sum_{l=0}^{L-1} y_k^{d+l}(t), \tag{18}$$

When using ISTFT, the parameters must be the same as those of STFT using Equation (2). If A is a constant related to the window function, $A = 0.5T/L$ when using Hanning window, and $y_k^d(t)$ is expressed as follows:

$$y_k^m(t) = \begin{cases} \sum_{f\in 0,\frac{1}{T}f_s,\ldots,\frac{T-1}{T}f_s} & Y_k(m,f)e^{j2\pi f r} \\ & mL \le t \le mL + T - 1, \\ 0 & \text{others} \end{cases}$$

where, $r = t - mL$.

2.3. Evaluation of Separation Performance

In order to verify the separation performance of the algorithm after adding noise, we simulated the binary time–frequency masking method. The T-F masking method requires the signal to meet the conditions of WDO or energy dominance. Therefore, the LFM signal was selected for the simulation to facilitate the aliasing operation of the signal for time and frequency. The detailed experimental process is described in Section 4. The experimental results show that when there is no noise, each signal can be recovered well, and the method can correctly divide the T-F region of each signal. Once noise is added, the performance deteriorates. The estimated masking matrix not only loses some of the information of the signal itself but also receives the T-F information of other signals.

3. Proposed Method

In recent years, deep learning has been successfully applied in speech separation [33,34], and these previous attempts have generally assumed that the numbers and types of sources are fixed. However, in the case of underwater acoustic signal separation, we must consider two problems: (1) the model can be used to separate arbitrary types of underwater acoustic sources, i.e., the generalization problem and (2) the model can be used to separate arbitrary numbers of underwater acoustic sources, i.e., the scalability problem. Unlike previous attempts, in this article we use deep learning methods to learn a mapping for the input that is amenable to clustering, and this is helpful in overcoming the above two shortcomings. The architecture of the proposed method is illustrated in Figure 2.

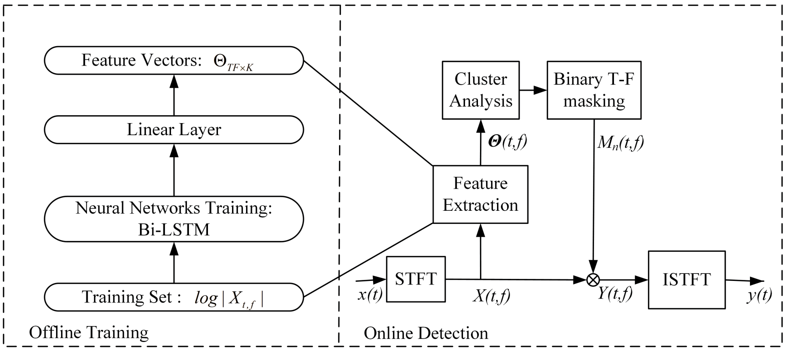


Figure 2. Framework of proposed approach.

Based on the traditional binary T-F masking method, this scheme uses the deep neural network to extract features from the original underwater acoustic data instead of using artificial feature extraction. The program is divided into two stages: offline training and online testing. (1) Offline training phase. The training of the network consists of three parts: STFT, preprocessing and network training. The data are obtained from the measured underwater sound database and preprocessed to obtain training samples. Then, the T-F map of the underwater acoustic signal is obtained, mainly through STFT. Fianlly, it is sent to the network for training. In order to ensure that the network learns from the original underwater acoustic characteristics to obtain cluster-oriented features, this paper sets an appropriate objective function to make the characteristics of the network output easier to cluster. (2) Online testing phase. The artificial feature extraction method in the traditional binary T-F masking method is replaced by the network with the previous stage's learning, and various mixed water acoustic signals are used to test whether the separation performance of the scheme meets the requirements. A flow chart for the specific method is shown in Figure 3.

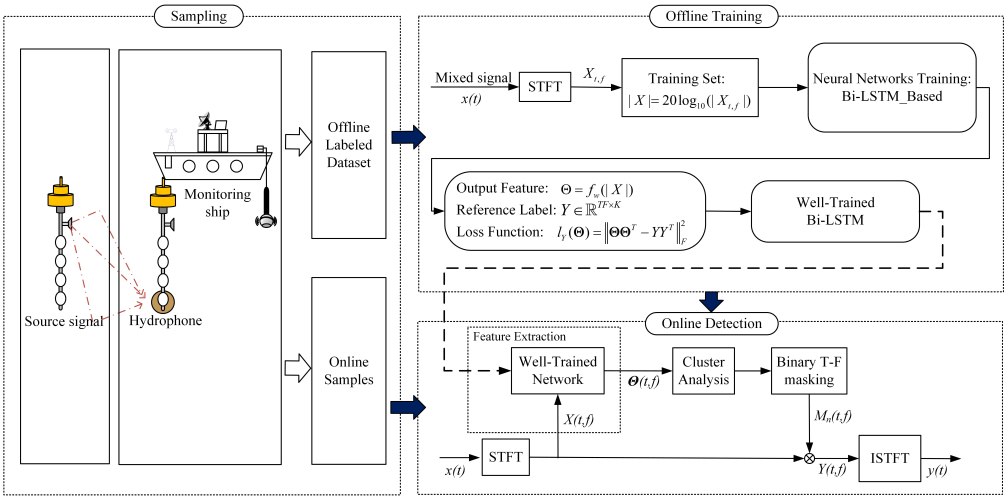


Figure 3. Flow chart of proposed method for underwater acoustic signal separation.

3.1. Feature Extraction Based on Deep Neural Network

In order to achieve good separation performance after clustering, it is required that the clustering features have good distinguishing characteristics. In recent years, many studies have used deep neural networks [35] to obtain powerful characterizations for clustering [20,29,36–42]. Good results have been achieved in image recognition and NLP. These approaches are characterized by embedding the original data features into the new feature space, making the transformed features more suitable for clustering. In addition to the target underwater acoustic signal, ship-radiated noise and marine environment noise also exist in the sonar system. Due to varying degrees of decay in the ocean, the main energy of these types of noise is concentrated at different frequencies. The main sound-source frequencies are shown in Table 1.

Table 1. Major ocean sound-source frequencies.

Sound Source	Frequency
Marine life	0.5–20 KHz
Radiated noise from ships	less than 1 kHz
Surface ships	100–500 Hz
Submarines	100–500 Hz

For a communication sonar receiving transmitted signals from other sonar platforms, the receiving bandwidth of the receiver is about 100 Hz to 3000 Hz, and the receiver has prior knowledge of these detection signals [20]. According to the characteristics of underwater acoustic signals, a neural network can be used to “divide” different types of signals in the water audio frequency domain, using a Fourier transform signal processing method to restore the signal. Finally the target signal can be separated. According to the embedding principle, the role of the deep neural network used in this section is to map the original features (immediate frequency features) of the measured data to the new feature space. Each T-F point is converted into a vector. Each vector has a different position in the new feature space, depending on the amount of energy at the T-F point. These vectors are then “divided” into a number of reasonable ranges based on the distance between the vectors. That is, the T-F vectors belonging to the same underwater sound source have similarities, such that the distance is the smallest, and the T-F vectors belonging to different underwater sound sources have a large distance. Finally, they can be easily divided using a simple clustering algorithm.

Suppose a mixed water acoustic signal is transformed by STFT to obtain the original T-F characteristic $X_{t,f} \in R^{T \times F}$, where t is the number of time frames and f is the frequency point. Taking the logarithmic amplitude spectrum $20\log10(|X_{t,f}|)$ as the input of the network, for convenience of description, the latter is uniformly recorded as $|X|$. $|X|$ can also be regarded as a sequence $[\chi_1, \chi_2, \ldots, \chi_T]$ composed of spectral information $\chi_i \in R^F$ over a plurality of consecutive times. The deep neural network is parameterized by ω, and the features generated based on the network are expressed as:

$$\Theta = f_\omega(|X|). \tag{19}$$

Here, $\Theta = [\theta_1, \theta_2, \ldots, \theta_{TF}]^T \in R^{TF\times K}$ is the whole amplitude information $|X|$ of the underwater acoustic signal, i.e., the cluster-oriented K-dimensional embedding feature learned by neural network. During the training process, the network sequentially maps the spectrum information χ_i to a new feature space at each time step and finally outputs it as an $F \times K$-dimensional vector. This can be considered as encoding each T-F point in the original T-F feature χ_i, and each T-F point after encoding is represented by a row vector θ_j of dimension K. Here θ_i is the unit vector, i.e., $|\theta_j|^2 = 1$.

The goal of training is to allow the line vector of the network output feature Θ to be divided into different water sources. That is, θ_j satisfies the vector distances belonging to the same water source, and the vectors belonging to different water sources are further away, thus achieving the purpose of separating the underwater sounds. Assuming that there is a mixed underwater sound in the water area, it is composed of C types of underwater sound sources:

$$x(t) = \alpha_1 s_1(t) + \alpha_2 s_2(t) + \ldots + \alpha_C s_C(t). \tag{20}$$

Before sending the mixed signals to train the network, the energy of each source signal is compared at each time and frequency point. First, we set the reference label $Y \in R^{TF\times C}$ to divide the time and frequency points and compare the energy of these C types of underwater sound sources at various time and frequency points. The energy-dominated underwater sound source will mark the time and frequency points. For example, if the energy of the c-th ($c \in \{1, 2, \ldots, C\}$) underwater sound, dominates at the n-th ($n \in \{1, 2, \ldots, TF\}$) time and frequency points, then $y_{n,c} = 1$. Therefore, the loss function of the model can be set as:

$$\begin{aligned} l_Y(\Theta) &= \|\Theta\Theta^T - YY^T\|_F^2 \\ &= \sum_{i,j} (\langle\theta_i, \theta_j\rangle - \langle y_i, y_j\rangle)^2 \\ &= \sum_{i,j:y_i=y_j} (|\theta_i - \theta_j|^2 - 1) + \sum_{i,j:y_i\neq y_j} \langle\theta_i, \theta_j\rangle^2, \end{aligned} \tag{21}$$

where $\| \bullet \|_F^2$ is the squared Frobenius norm [43]. In the process of minimizing the loss function, two vectors for the same water source will become closer and closer, and the distance between two vectors for different water sources will increase. At the same time, since $(YP)(YP)^T = YY^T$ exists for any permutation matrix P, the method can ensure that the label arrangement and the number of all training samples are independent.

3.2. *Offline Training: Test Network Design Based on RNN, LSTM and Bi-LSTM, Respectively*

Input and reference label processing: First, randomly take (2 C) underwater acoustic audio files from the file library and mix them according to Equation (20). Each audio file must be averaged before entering the network training stage:

$$s'(t) = s(t) - E[s(t)], \tag{22}$$

$$s''(t) = \frac{s'(t)}{max(|s'(t)|)}. \tag{23}$$

The mixing coefficient α is randomly taken as an arbitrary number in the 16 s interval [3/4, 1]. According to Equation (2), the mixed signal has a window length of 32 ms and a time shift STFT of 8 ms, and the log amplitude spectrum X is taken. For a 16 s audio, it can be split into 500 samples of size 706. At the same time, we take the logarithmic amplitude spectrum of each source signal that makes up the mixed signal and compare the magnitude of the energy at each time and frequency point, to form the reference label Y with the same shape as X. To ensure local accuracy, each iteration consists of a sequence of time steps from multiple input samples of X and Y, and each sequence is 50% overlapped, to form a minimum batch-pair network for training.

In the offline training phase, in order to more clearly introduce the proposed Bi-LSTM structure used in this paper and highlight its superiority compared with other neural networks, we tested three structures: RNN, LSTM and Bi-LSTM. In addition, since LSTM is closely related to Bi-LSTM, the following section will first give a brief description of the LSTM structure, followed by a detailed introduction to Bi-LSTM.

Structure 1 (LSTM-based): RNN has long-term dependency problems. As the structural model of RNN becomes deeper, RNN must repeatedly apply the same operations to each moment in the long-term sequence to generate a very deep computational graph. Coupled with model parameter sharing, RNN is prone to losing the ability to learn previous information, making optimization extremely difficult. Unlike RNN's regular loop body structure, LSTM uses neurons dedicated to memory storage. The neuron is a special network structure with three "gate" structures, called input gates, output gates and forgetting gates. During training, the LSTM relies on these gated operations (reset and read and write operations) to selectively influence the state of each moment in the network. After the investigation, we know that feature extraction can be performed using RNN. However, we use LSTM networks in this study, which is an improvement on RNN [44].

LSTM can form a deep LSTM network by stacking, repeating the loop body at each moment to enhance the expressive ability of the model. The parameters of the loop body of each layer are the same, and the loop body parameters of different levels can be different. A schematic diagram of the network structure for water acoustic separation using multilayer LSTM is shown in Figure 4. By stacking, the neural network can learn deeper expressions and finally embed them into the K-dimensional features.

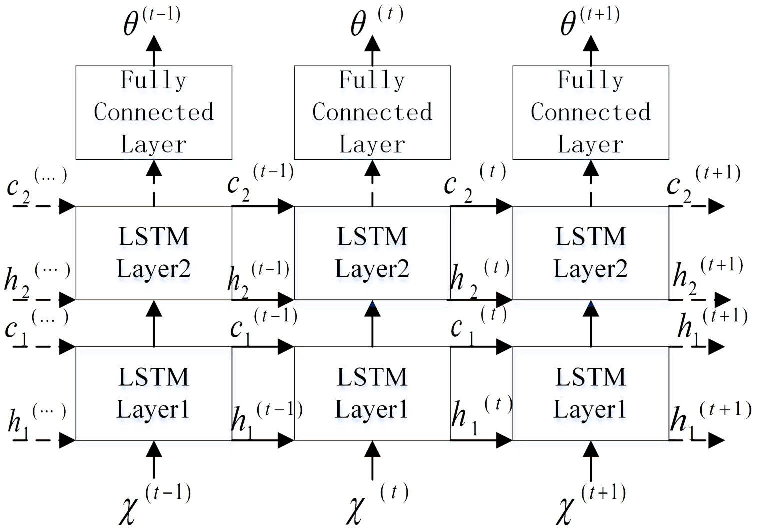


Figure 4. LSTM underwater acoustic separation network structure.

Structure 2 (Bi-LSTM-based): The transmission of the two network structures, RNN and LSTM, is one-way from front to back, that is, the state at time t can only capture information from the past sequence $x^1, \dots, x^{t-1}$ and the current input x^t. For some problems, however, the prediction of the output may depend on the entire sequence. For example, in speech recognition, some words currently have multiple interpretations and must be judged in context. Therefore, the processing of the voice must refer to the pronunciation information in the past and the future in order to produce a more accurate effect.

It is also possible to encounter the same problem in the field of underwater sound. For example, in underwater acoustic communication, sound waves are used instead of radio waves, due to the serious attenuation of underwater waves. Therefore, in underwater communication, the transmission of text, voice, images and other information needs to be converted into an electrical signal and then converted into an acoustic signal. At this time, in order to separate the speech signal in the water from noise such as waves, fish and ships, the influence of the front and back states on the output should be considered. During the collection of, and research into, marine sounds, the sound of fish as a signal for communication between fish schools should also consider the impact of the entire sequence on the output of the network. To this end, Bi-LSTM can be used to make full use of the context information in the sample for training.

Bi-LSTM consists of two LSTMs of the same size and opposite starting points of the time series. Figure 5 shows the structure of a water acoustic separation network based on Bi-LSTM. Here, $h^{(t)}$ represents the state of the sub-LSTM that propagates information from $t = 1$ to T (to the right) in time and $h'^{(t)}$ represents the state of the sub-LSTM in which the information moves backward from $t = T$ to 1 (to the left) and can be obtained by substituting the reverse sequence into Equations (24)–(28). The specific operation of the unidirectional sub-LSTM layer is as follows. Given an input sequence $\mathbf{X} = \{X_1, \ldots, X_T\}$, this model can be iteratively computed from $t = 1$ to T and is composed of the following:

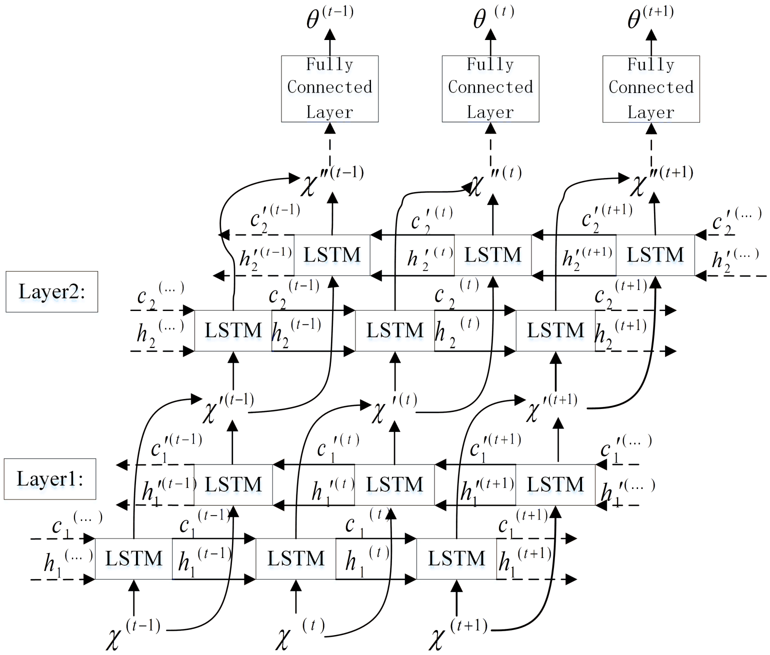


Figure 5. Bi-LSTM underwater acoustic separation network model diagram.

$$i_t = \sigma(W_{Xi}X_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \tag{24}$$

$$f_t = \sigma(W_{Xf}X_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \tag{25}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{Xc}X_t + W_{hc}h_{t-1} + b_c), \tag{26}$$

$$o_t = \sigma(W_{Xo}X_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \tag{27}$$

$$h_t = o_t tanh(c_t), \tag{28}$$

where W and b are weights and biases, and i, f, o and c are the input gate, forget gate, output gate and cell activation vector, respectively. In addition, σ is the logistic sigmoid function. Therefore, at each time point t, the output unit can obtain information about the past sequence with respect to the input $h'^{(t)}$ and the relevant information about the future sequence of the input $h^{(t)}$. After the two sub-LSTM layers, we use a dense layer to obtain Θ_t, which is the output of the X_i:

$$\Theta_t = \phi(W_h h_t^l + b_\Theta), \tag{29}$$

where h_t^l is the output of the final LSTM layer and ϕ is the ReLU activation function. By minimizing the loss value, some parameters will adaptively change as the learning process advances.

In the following experiments, we extracted the characteristics of the underwater acoustic signal using the above three networks (RNN, LSTM and Bi-LSTM) in the offline training phase. In the online test phase, combined with STFT and binary time–frequency masking methods, we obtained the corresponding experimental data for the three networks. The experiments showed that the Bi-LSTM structure had the best performance. However, the proposed scheme cannot handle the situation with unknown signal waveforms. It should be noted that our model can only handle known signals such as test signals for calibration; however, this is not an unreasonable constraint [45].

3.3. Online Test

Different models were trained and applied to the traditional binary T-F masking framework. The processing flow of the method is basically the same as the processing flow of the binary T-F masking method. The main steps are as follows:

(1) Select the underwater acoustic signal in the test set for mixing to obtain a mixed underwater acoustic signal. The signal is de-equalized and normalized, and the signal is subjected to STFT (the parameters of the STFT in the test phase are consistent with the STFT parameters in the training phase). Finally, $|X|$ is obtained as an input.
(2) Using the trained model, the original feature X of the signal is transformed into a new embedded feature Θ. Since the new feature is just a matrix of dimension $T \times FK$ when it is output from the network, in the actual processing, we must reshape the data, convert their dimensions to $TF \times K$, and facilitate the subsequent cluster analysis.
(3) Cluster analysis. The clustering analysis of feature Θ is performed using the K-means algorithm.
(4) T-F masking. According to the set Ω_k obtained by clustering, the corresponding binary T-F masking matrix $M_k(t, f)$ is set and substituted into Equation (17), thereby obtaining a T-F-domain estimate of the source signal.
(5) Time-domain recovery. The source signal $\tilde{S_k}(t, f)$ estimated in the above step is subjected to ISTFT estimation according to Equation (18), to obtain the time-domain waveform $\tilde{s_k}(t)$ of the source signal.

The clustering algorithm is used to classify this feature of the neural network output such that the vectors θ belonging to the same underwater sound source can be divided into a group. We set each "similar" vector to 1 and set the vectors that are not similar to 0. The new array dimension is reconstructed into a $T \times F$ matrix, which is the binary masking matrix corresponding to the water source.

4. Experiments

4.1. Experimental Conditions

For the experiments, we selected a hydroacoustic audio dataset in ShipsEar as a data sample [46]. Since its establishment, the database has been used for research on ship noise reduction, detection, identification, etc., especially for the application of deep learning technology [47–49]. The hydroacoustic data in this database were collected by the researcher David, a hydrologist from the Atlantic coast of northwestern Spain, and others from the University of Vigo in Spain. The composition of the database is shown in Table 2. The sonar audio, ship radiation noise and background noise form the A, B and C signals, respectively, and each audio file was selected to be about 6 seconds in length for testing. The sampling rate was unified to 44,100 Hz. In addition, we also simulated the binary time–frequency masking method. By comparing the effects of binary separation and multiple separation, the superior performance of the proposed method was proved. For the binary time–frequency masking method, we selected three LFM signals for simulation, which facilitates the aliasing operation of the signals for time and frequency. For the three LFM signals simulated, the sampling frequency was 50 kHz and the time length was 1 s. The specific parameters are shown in Table 3.

Table 2. The composition of the database.

Category	Details
Number of recordings	90 segments
Recording length	15 s to 10 min
Number of ships	11
Background noise	Different depths and channel distances

Table 3. LFM signal parameters for binary time–frequency masking method simulation.

Signals	Frequency Range	Launch Time	Duration
LFM 1	6–8 kHz	0.1 s	0.3 s
LFM 2	6.5–10 kHz	0.5 s	0.2 s
LFM 3	12–15 kHz	0.6 s	0.3 s

In the training stage, we attempted to train the model with a maximum mixture number of three. Hence, we randomly selected two or three files from the training set to mix in every iteration. Then, we used the model to separate each possible underwater acoustic mixing source. We designed the network structure with two LSTM layers with 600 hidden cells and a full connection layer with 100 cells, corresponding with the embedding dimension K. Stochastic gradient descent with momentum 0.9 and a fixed learning rate of 10^{-5} was used for training. The ReLU function was used as the activation function for the output layer, with order n. To prevent the network from overfitting and improve the generalization ability of the model, the input layer and the hidden layer's dropout parameters were set to 0.2 and 0.5, respectively. When adding L2 regularization to the network, the parameter was set to 10^{-6}. The number of training iterations of the model was 30.

In the test stage, the input feature X was the log magnitude spectrum of the mixed underwater acoustic signal, using STFT with 32 ms frame length, 8 ms window shift and the square root of the Hanning window. Moreover, the mixture was separated into 100 frames with half overlap to ensure the local accuracy of the output feature Θ. The masks were obtained by clustering the row vectors of the feature Θ. The number of clusters was set to the number of sources in the mixture.

4.2. Metrics

To evaluate the quality of the source separation, we used three quantitative criteria: (1) the preserved-signal ratio (PSR$\in [0,1]$), representing the quality of the mask preserving the target source; (2) the signal-to-interference ratio (SIR$\in [0,\infty)$), representing the quality of the mask suppressing the interfering sources; and (3) the similarity coefficient ξ, which estimates the similarity between the signal $y_i(t)$ and the source signal $x_j(t)$.

PSR: The preserved-signal ratio (PSR) is used to measure the degree of protection of the masking matrix M_k from the target signal $X_k(t,f)$. The mathematical equation is expressed as follows:

$$PSR = \frac{\|M_k(t,f)X_k(t,f)\|^2}{\|X_k(t,f)\|^2}, \tag{30}$$

The PSR characterizes the amount of energy remaining after the target signal passes through the masking matrix. In the equation, $\|\cdot\|^2$ represents a double integral operation, that is, $\|f(x,y)\|^2 = \iint |f(x,y)|^2 dxdy$. The PSR satisfies $0 \le PSR \le 1$. If the estimated masking matrix M_k satisfies the relationship $\hat{M}_k \subseteq M_k$ with the actual masking matrix $\hat{M}_k$, then PSR = 1.

SIR: The SIR indicates the suppression of the interference source by the masking matrix. An interference source composed of source signals other than the source signal $x_k(t)$ is denoted by $v_k(t)$, and the corresponding T-F domain is expressed as $V_k(x, y)$. The signal-to-interference ratio for the masking matrix M is defined as follows:

$$SIR_M = \frac{\|M_k(t, f)X_k(t, f)\|^2}{\|M_k(t, f)V_k(t, f)\|^2}, \tag{31}$$

where SIR_M is a value greater than or equal to 0. The larger the value, the better the separation performance. When the masking matrix is completely suppressed with respect to the other source signals, $SIR_M = \infty$. In the T-F mask separation method, good separation performance requires that the T-F information of the source signal is preserved as much as possible and that the interference source can be suppressed, that is, the PSR is close to 1 and SIR_M is as large as possible.

ξ**:** The similarity coefficient ξ is given by

$$\xi_{ij} = \xi(y_i, x_j) = \frac{\left|\sum_{t=1}^{n} y_i(t)x_j(t)\right|}{\sqrt{\sum_{t=1}^{n} y_i^2(t) \sum_{t=1}^{m} x_j^2(t)}}. \tag{32}$$

If $\xi_{ij} = 1$, this means that the i-th estimated signal is exactly the same as the j-th source signal. If $\xi_{ij} = 0$, this means that $y_i(t)$ and $s_j(t)$ are completely inconsistent. In an actual situation, due to the existence of the estimated difference, the separation performance of the similarity coefficient is generally close to 1, and the worst value is 0. Generally, these coefficients constitute a similarity coefficient matrix. If only one similarity coefficient in each row in the matrix tends to 1 and the others tend to 0, the separation performance is good.

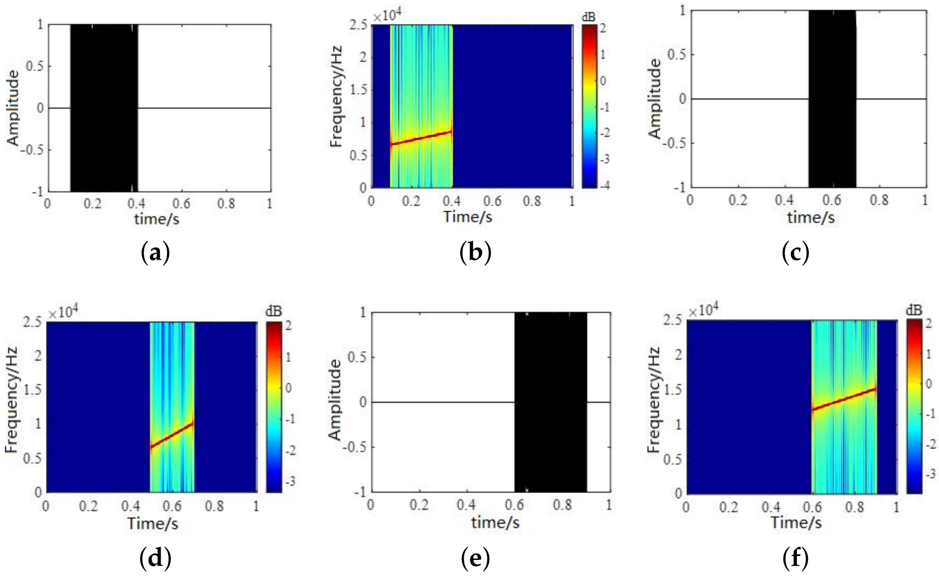
4.3. Results

4.3.1. Binary Signal Separation Using Binary Time–Frequency Masking Method

In the experiments, we first simulated three LFM signals that satisfy the energy-dominated condition. The time-domain waveforms and time–frequency diagrams of the simulation signals are shown in Figure 6. Figure 6a,c,e show the time-domain waveforms of the three signals, and Figure 6b,d,f show the time–frequency diagrams for the three signals.

The randomly generated mixing matrix is linearly mixed according to Equation (33). In Equation (33), A represents the underwater channel matrix and n(t) represents white noise. The time-domain waveforms and time–frequency diagrams of the observed signals are shown in Figure 7. We selected the Hamming window for the observation signal, performed a 512-point STFT transformation and set a 25% overlap to obtain the time–frequency characteristics. According to Equations (7) and (9), we took the magnitude of the observation signal and the phase difference to form a feature vector, and finally obtained an estimated signal. The time-domain waveforms and time–frequency diagrams of the observed and estimated signals are shown in Figure 7. It can be seen from the results that when the signal meets the sparsity condition, the binary signal can be recovered using a binary time–frequency masking algorithm. The time-domain waveform and time–frequency diagram for the estimated signal are shown in Figure 8. From the effect diagram of the estimated signal, the source signal can be basically recovered using the binary time–frequency masking method. Source signal 2 is aliased with source signals 1 and 3 in the frequency domain and time domain, respectively, and so the information will be somewhat affected but can basically be recovered from the mixed signal.

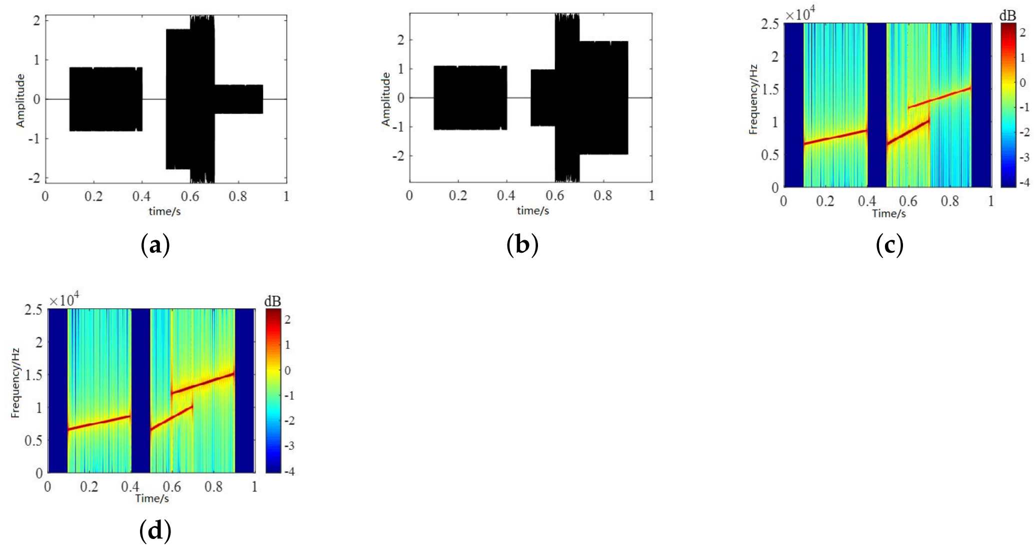
$$x(t) = As(t) + n(t). \tag{33}$$



(a) (b) (c)

(d) (e) (f)

Figure 6. Time-domain waveforms and time–frequency diagrams for source signals. (**a**,**c**,**e**) show the time-domain waveforms of the three signals and (**b**,**d**,**f**) show the time–frequency diagrams for the three signals.



(a) (b) (c)

(d)

Figure 7. Time-domain waveforms and time–frequency diagrams of observed signals. (**a**,**b**) are the time-domain waveforms diagrams of the observed signals and (**c**,**d**) are the time-frequency diagrams.

The correlation coefficients ξ and the PSR and SIRM were measured under different signal-to-noise ratios. The results are shown in Table 4. It can be seen that when there is no noise, each signal can be recovered well. The two parameters PSR and SIR indicate that the method can correctly divide the time–frequency region of each signal, that is, the obtained masking matrix accurately covers the time of the signal and frequency information. Once noise is added, the performance deteriorates. The PSR reduction is small, but the SIRM reduction is obvious. This means that after adding noise, the estimated masking matrix not only loses some of the information of the signal itself but also receives time–frequency information from other signals.

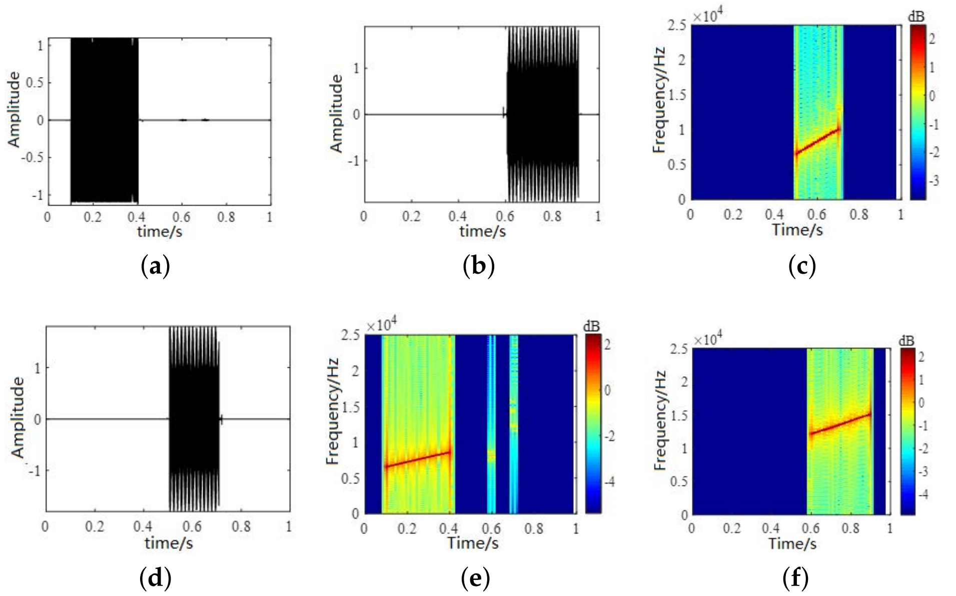


Figure 8. Time-domain waveforms and time–frequency diagrams of estimated signals. (**a**,**b**,**d**) show the time-domain waveforms of estimated signals and (**c**,**e**,**f**) show the time–frequency diagrams.

Table 4. Separation performance at different SNRs.

SNR/dB	0	5	10	15	20	No Noise
ξ	0.60	0.62	0.74	0.81	0.89	0.98
PSR	0.71	0.72	0.82	0.85	0.90	0.98
SIR_M	5.82	5.47	15.56	27.93	316.21	24,193.72

4.3.2. Binary and Multivariate Signal Separation Using the Proposed Method

Next, we separated the mixed signals with two sources. The visualization of the result can be seen in Figure 9. We listed all possible combinations and observed the corresponding effect on separation. Figure 9a,c,e show the spectrum of sources A, B and C separately. Figure 9b,d,f show the separation results of the pairwise mixtures of A, B and C, respectively. Compared with the original spectra, it can be seen that the Bi-LSTM model can clearly separate the A, B and C signals before mixing.

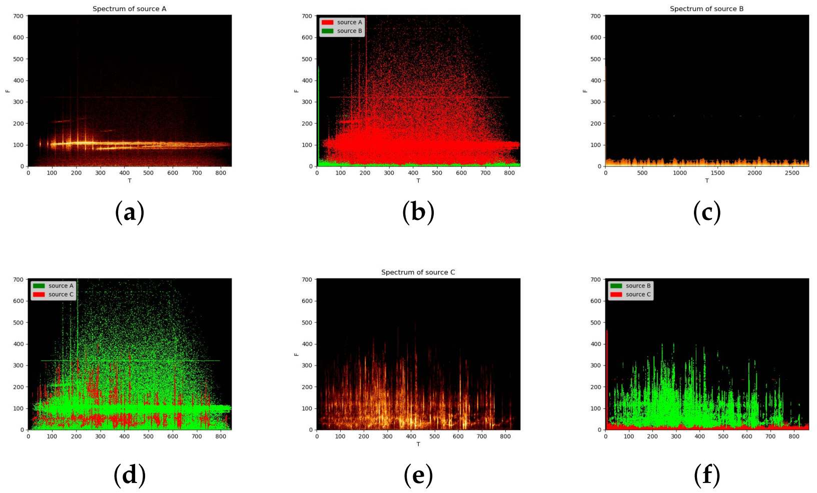


Figure 9. Visualization of separation of two-source mixtures. (**a**,**c**,**e**) show the spectrum of sources A, B and C separately and (**b**,**d**,**f**) show the separation results of the pairwise mixtures of A, B and C, respectively.

In Table 5, we illustrate the de-mixing performance for separating two sources using the metrics mentioned in Equations (30) and (31). This shows that our proposed method had better performance in separating two sources, which indicates that this approach is different from many separation algorithms based on deep learning. SIR is infinity because the interfering sources are suppressed sufficiently, making the denominator close to 0 according to Equation (31).

Table 5. The de-mixing performance in the experiments.

Sources	SIR in (dB)	SIR out (dB)	SIR Gain (dB)	PSR
Source A	−14.28	∞	∞	0.93
Source B	14.28	∞	∞	0.92
Source C	−13.74	∞	∞	0.90

Furthermore, we separated mixed signals from three sources. Figures 10 and 11 show an example of separating the mixtures of three sources. By comparing Figures 9 and 10, it can be seen that the time and frequency points of each source can basically be found. The overlap between source signal C and source signal A is relatively large in the time–frequency domain. However, signal A dominates with respect to the energy at these overlapped time and frequency points, so it will not be disturbed by the signals and can basically be recovered. However, some information in signal C is lost. In fact, compared with background noise, people are more concerned about the loss of a sonar echo signal. Therefore, it is permissible to sacrifice part of signal C in practical applications. The overlap between signal B and signals A and C in the frequency domain is the least, and the separation performance is the best. However, in order to prove that using a deep learning method to separate underwater acoustic sources can achieve a breakthrough, we also show the results using the traditional binary T-F mask approach. In Table 6, the first example is our approach, and the second is the traditional approach. It is clear that our proposed method outperforms the traditional method, which cannot even separate sources C and A very well. What is more, compared with Table 5, when we separate more sources, the performance does not decrease too much. Therefore, the proposed model can be scaled up to more sources. Thus, it is appropriate for real-world applications when the number of sources is not fixed.

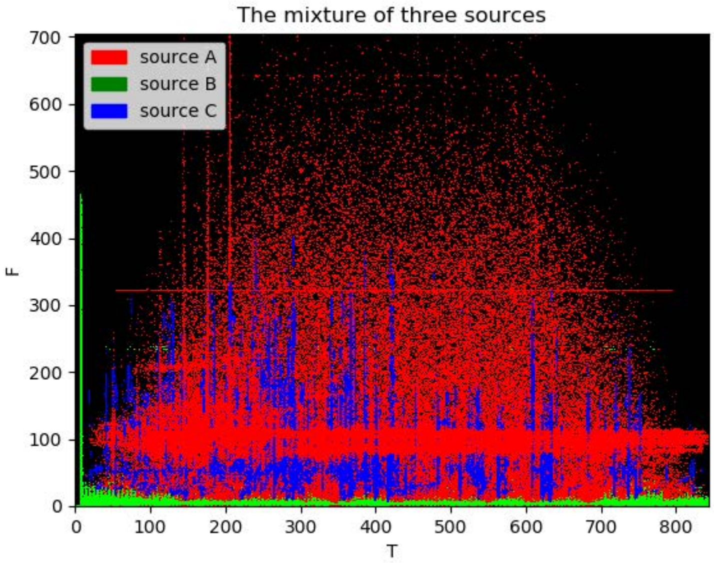


Figure 10. Separation results for three-source mixtures.

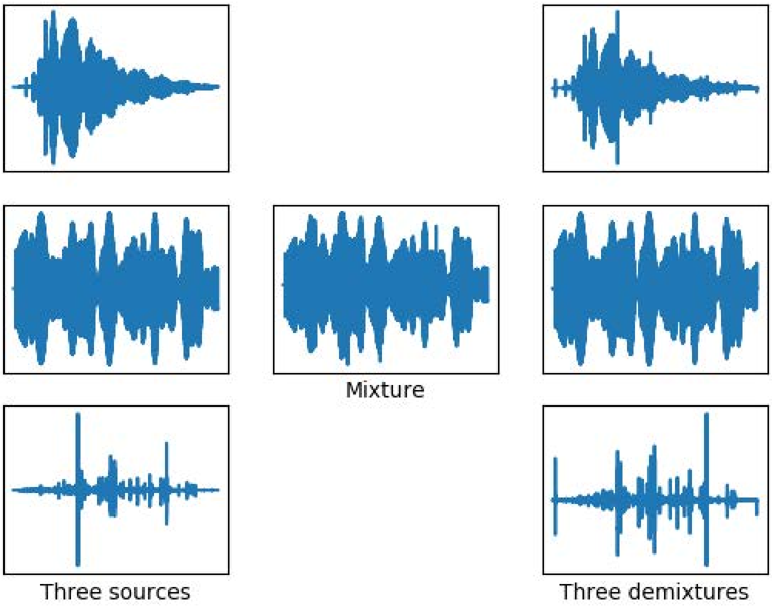


Figure 11. Three sources, the mixture and three de-mixed signals.

Table 6. Comparison of the de-mixing performance in Experiment 3 (top) with that of the conventional T-F mask approach (bottom).

Sources	SIR in (dB)	SIR out (dB)	SIR Gain (dB)	PSR
Source A	−14.29	∞	∞	0.94
Source B	14.10	∞	∞	0.93
Source C	−28.18	∞	∞	0.90
Source A	−14.29	13.93	28.22	0.81
Source B	14.10	∞	∞	0.93
Source C	−28.18	2.42	30.6	0.29

In addition, considering that the mixed signal will be subject to interference from other unknown noises in the actual processing, Gaussian noise signals of 0–40 dB were added to the mixed signal, to analyze the separation performance under different SNR conditions. Meanwhile, compared with the traditional binary T-F masking method, the similarity coefficient was used as the measurement standard. The results are shown in Figure 12. When the noise background is relatively strong, both the deep-learning-based separation method and the traditional separation method will be greatly affected. As the noise is reduced, the estimated signal gradually becomes clear. Compared with the above separation situation, this test signal has larger aliasing in both the frequency domain and the time domain. Therefore, the traditional time–frequency masking method has a poor separation performance, and its final average similarity coefficient is stable at about 0.6. The deep-learning-based separation method can divide each target signal according to the energy-dominant condition, and therefore it has better separation performance on the whole.

Under the condition of unknown noise, the separated signal will still carry noise, affecting the performance. It was found that the noise could be separated as long as the number of clusters was increased when the clustering algorithm was used. Taking the case of adding Gaussian white noise with an SNR of 0dB as an example, when the signal is divided into three categories, each signal will carry noise. Among them, signal C suffers the largest interference and has a very weak energy, as shown in Figure 13a. By increasing the number of clusters to four, that is, setting the K value of the K-means clustering algorithm to four, the proposed method could also separate noise from three source signals and recover the basic shape of signal C, as shown in Figure 13b. It can be proved that the proposed method can not only perform well in the separation of multivariate signals but can also work effectively in the presence of certain noise interference.

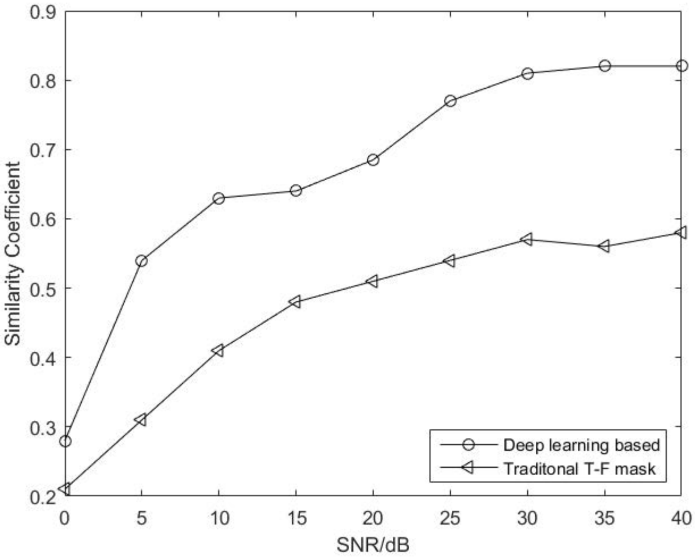


Figure 12. Comparison of similarity coefficients under different noise backgrounds.

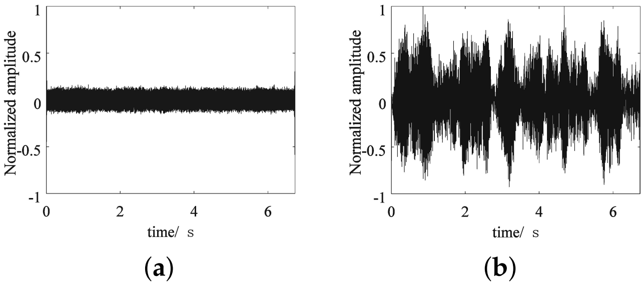


Figure 13. The separation results for signal C when different K values are set: (**a**) K = 3; (**b**) K = 4.

Finally, three models, RNN, LSTM and Bi-LSTM, were selected for comparison. Each model separated the mixed signals composed of sources A, B and C. The similarity coefficient ξ, and PSR and SIR_M were selected as comparison indicators, and the comparison results are shown in Table 7.

Table 7. Comparison of RNN, LSTM and Bi-LSTM models.

	A			B			C		
	ξ	PSR	SIR_M	ξ	PSR	SIR_M	ξ	PSR	SIR_M
RNN	0.43	0.95	1.84	0.44	0.99	27.84	0.21	0.63	0.66
LSTM	0.91	0.96	91.99	0.82	0.94	205.88	0.71	0.73	8.74
Bi-LSTM	0.92	0.99	111.79	0.93	0.97	10576	0.77	0.76	8.68

According to the results of Table 7, RNN performed the worst. Although the PSR of the A and B signals reached 0.99, SIR_M is very low. This shows that although the T-F information of the source signal was preserved, most of the T-F points that were not part of the source signal were also classified as source signals. The recovered signal then contains other signal components. In addition to retaining the original information well, LSTM and Bi-LSTM can also implement interference suppression for other signals. Bi-LSTM has a better suppression effect than LSTM. Signal B has the best recovery of the three configurations, especially in Bi-LSTM where the SIR_M reached 10,576 and the PSR reached

0.97. Comparing the distribution of the three sources, it can be seen that signal B and signals A and C have almost no overlap in the frequency domain, and hence they are easily distinguished.

5. Conclusions

In this paper, a deep learning separation method for underwater acoustic signals based on the T-F mask method was proposed. The method mainly uses Bi-LSTM to create the features of the time–frequency mask for clustering. In this way, each T-F bin is "encoded" directly and partitioned into a reasonable region according to its magnitude. For real-world tasks, it is important for the proposed model to have good scalability, since the number of target sources is not fixed. At the same time, the model should have good generalization ability, so that it can work effectively when separating uncertain underwater acoustic mixed sources in online applications. In order to illustrate the universality and extensibility of the model, we conducted experiments on two unknown mixed sources and three mixed sources, respectively, and tested the robustness of the model by adding 0–40 dB Gaussian noise. Finally, we compared and analyzed the performances of the RNN, LSTM and Bi-LSTM networks in extracting underwater acoustic signal characteristics. The results showed that the proposed method could obtain better performance under the conditions of large mixed-signal uncertainty and large Gaussian noise, showing an obvious improvement compared with the traditional T-F mask method. The most important point is that compared with mainstream methods, this model not only has better separation performance for binary signal separation but can also effectively separate aliased signals in the case of multiple signal separation, which cannot be handled well by existing methods.

Author Contributions: Conceptualization, J.C., C.L. and J.X.; Writing—original draft, J.A.; Writing—review & editing, N.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the China National Key R&D Program under grant 2020YFB1807700.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Chen, J.; Han, B.; Ma, F.; Zhang, J. Underwater target recognition based on multi-decision lofar spectrum enhancement: A deep-learning approach. *Future Internet* **2021**, *13*, 265. [CrossRef]
2. Kim, Y.G.; Kim, Y.; Lee, S.H. Underwater acoustic sensor fault detection for passive sonar systems. In Proceedings of the 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE), Aalborg, Denmark, 6–8 July 2016; pp. 1–4.
3. Liu, S.; Ma, T.; Qiao, G.; Ma, L.; Yin, Y. Biologically inspired covert underwater acoustic communication by mimicking dolphin whistles. *Appl. Acoust.* **2017**, *120*, 120–128. [CrossRef]
4. Bereketli, A.; Bilgen, S. Remotely powered underwater acoustic sensor networks. *IEEE Sens. J.* **2012**, *12*, 3467–3472. [CrossRef]
5. Rahmati, M.; Pompili, D. Unisec: Inspection, separation, and classification of underwater acoustic noise point sources. *IEEE J. Ocean. Eng.* **2017**, *43*, 777–791. [CrossRef]
6. Cardoso, J.F.; Comon, P. Independent component analysis, a survey of some algebraic methods. In Proceedings of the 1996 IEEE International Symposium on Circuits and Systems, Circuits and Systems Connecting the World, ISCAS 96, Atlanta, GA, USA, 15 May 1996; pp. 93–96.
7. Wang, D.; Zhao, A.; Hui, J. Research on acoustic three-user communication based on single vector sensor. In Proceedings of the 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 24–26 September 2009; pp. 1–4.
8. Cardoso, J.F. Blind signal separation: Statistical principles. *Proc. IEEE* **1998**, *86*, 2009–2025. [CrossRef]
9. Lee, D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2000**, *13*, 556–562.
10. Comon, P.; Jutten, C. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*; Academic Press: Pittsburgh, PA, USA, 2010.

11. He, H.; Cang, Y. The application research of underwater acoustic sourcenumbers estimation by blind separation algorithm. In Proceedings of the 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 11–13 December 2009; pp. 1–4.
12. Kirsteins, I.P. Blind separation of signal and multipath interference for synthetic aperture sonar. Oceans 2003. In Proceedings of the Celebrating the Past… Teaming Toward the Future (IEEE Cat. No.03CH37492), San Diego, CA, USA, 22–26 September 2003; pp. 1–5.
13. Heli, H.; Abutalebi, H.R. Localization of multiple simultaneous sound sources in reverberant conditions using blind source separation methods. In Proceedings of the 2011 International Symposium on Artificial Intelligence and Signal Processing (AISP), Tehran, Iran, 15–16 June 2011; pp. 1–5.
14. Ozerov, A.; Févotte, C. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *18*, 550–563. [CrossRef]
15. Comon, P. Independent component analysis, a new concept? *Signal Process.* **1994**, *36*, 287–314. [CrossRef]
16. Jourjine, A.; Rickard, S.; Yilmaz, O. Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. In Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (Cat. No. 00CH37100), Istanbul, Turkey, 5–9 June 2000; pp. 2985–2988.
17. Yilmaz, O.; Rickard, S. Blind separation of speech mixtures via time-frequency masking. *Signal Process. IEEE Trans.* **2004**, *52*, 1830–1847. [CrossRef]
18. Araki, S.; Sawada, H.; Mukai, R.; Makino, S. Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors. *Signal Process.* **2007**, *87*, 1833–1847. [CrossRef]
19. Araki, S.; Makino, S.; Blin, A. Underdetermined blind separation for speech in real environments with sparseness and ica. In Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, QC, Canada, 17–21 May 2004; pp. 1–5.
20. Tian, F.; Gao, B.; Cui, Q. Learning deep representations for graph clustering. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 21 June 2014; pp. 1–5.
21. Liu, X.; She, C.; Li, Y. Edge-wise gated graph neural network for user association in massive urllc. In Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps), Madrid, Spain, 7–11 December 2021; pp. 1–6.
22. Liu, X.; Liu, C.; Li, Y.; Vucetic, B.; Ng, D.W.K. Deep residual learning-assisted channel estimation in ambient backscatter communications. *IEEE Wirel. Commun. Lett.* **2020**, *10*, 339–343. [CrossRef]
23. Liu, C.; Liu, X.; Ng, D.W.K.; Yuan, J. Deep residual learning for channel estimation in intelligent reflecting surface-assisted multi-user communications. *IEEE Trans. Wirel. Commun.* **2021**, *21*, 898–912. [CrossRef]
24. Li, Y.; Wang, D. On the optimality of ideal binary time–frequency masks. *Speech Commun.* **2009**, *51*, 230–239. [CrossRef]
25. Sawada, H.; Araki, S.; Mukai, R.; Makino, S. Blind extraction of dominant target sources using ICA and time-frequency masking. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 2165–2173. [CrossRef]
26. Roman, N.; Wang, D.L.; Brown, G.J. Speech segregation based on sound localization. *J. Acoust. Soc. Am.* **2003**, *114*, 2236. [CrossRef]
27. Rickard, S.; Balan, R.; Rosca, J. Real-time time-frequency based blindsource separation. *AJE* **2001**, *2*, 1.
28. Liu, C.; Wang, J.; Liu, X.; Liang, Y.C. Maximum eigenvalue-based goodness-of-fit detection for spectrum sensing in cognitive radio. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7747–7760. [CrossRef]
29. Xie, J.; Fang, J.; Liu, C.; Li, X. Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach. *IEEE Commun. Lett.* **2020**, *24*, 2196–2200. [CrossRef]
30. Liu, C.; Li, M.; Jin, M.L. Blind energy-based detection for spatial spectrum sensing. *IEEE Wirel. Commun. Lett.* **2014**, *4*, 98–101. [CrossRef]
31. Liu, C.; Li, H.; Jin, M. Blind central-symmetry-based feature detection for spatial spectrum sensing. *IEEE Trans. Veh. Technol.* **2016**, *65*, 10147–10152. [CrossRef]
32. Raki, S.; Makino, S.; Sawada, H. Reducing musical noise by a fine-shift overlap-add method applied to source separation using a time-frequency mask. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'05), 2005, Philadelphia, PA, USA, 23 March 2005; pp. 1–81.
33. Weninger, F.; Eyben, F.; Schuller, B. Single-channel speech separation with memory-enhanced recurrent neural networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 1–5.
34. Huang, P.S.; Kim, M.; Hasegawa-Johnson, M.; Smaragdis, P. Deep learning for monaural speech separation. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 1–5.
35. Liu, C.; Yuan, W.; Li, S.; Liu, X.; Li, H.; Ng, D.W.K.; Li, Y. Learning-based Predictive Beamforming for Integrated Sensing and Communication in Vehicular Networks. *IEEE J. Sel. Areas Commun.* **2022**, *10*, 22–24. [CrossRef]
36. Xie, J.; Fang, J.; Liu, C.; Yang, L. Unsupervised deep spectrum sensing: A variational auto-encoder based approach. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5307–5319. [CrossRef]
37. Huang, P.; Huang, Y.; Wang, W. Deep embedding network for clustering. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 1532–1537.

38. Mikolov, T.; Sutskever, I.; Chen, K. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 21 December 2013; pp. 3111–3119.
39. Song, C.; Liu, F.; Huang, Y. Auto-encoder based data clustering. *Iberoam. Congr. Pattern Recognit.* **2013**, *3*, 117–124.
40. Alqahtani, A.; Xie, X.; Deng, J. A deep convolutional auto-encoder with embedded clustering. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 4058–4062.
41. Liu, C.; Wei, Z.; Ng, D.W.K.; Yuan, J.; Liang, Y.C. Deep transfer learning for signal detection in ambient backscatter communications. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 1624–1638. [CrossRef]
42. Liu, C.; Wang, J.; Liu, X.; Liang, Y.C. Deep CM-CNN for spectrum sensing in cognitive radio. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2306–2321. [CrossRef]
43. Hershey, J.R.; Chen, Z.; Le Roux, J.; Watanabe, S. Deep clustering: Discriminative embeddings for segmentation and separation. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 31–35.
44. Gers, F.A.; Schraudolph, N.N.; Schmidhuber, J. Learning precise timing with lstm recurrent networks. *J. Mach. Learn. Res.* **2002**, *3*, 115–143.
45. Xu, W.; Chen, C.; Dai, J.; Zhou, Y. Detection of known signals in additive impulsive noise based on Spearman's rho and Kendall's tau. *Signal Process.* **2019**, *161*, 165–179. [CrossRef]
46. Santos-Domínguez, D.; Torres-Guijarro, S.; Cardenal-López, A.; Pena-Gimenez, A. Shipsear: An underwater vessel noise database. *Appl. Acoust.* **2016**, *113*, 64–69. [CrossRef]
47. Shen, S.; Yang, H.; Sheng, M. Compression of a deep competitive network based on mutual information for underwater acoustic targets recognition. *Entropy* **2018**, *20*, 243. [CrossRef]
48. Shen, S.; Yang, H.; Li, J.; Xu, G.; Sheng, M. Auditory inspired convolutional neural networks for ship type classification with raw hydrophone data. *Entropy* **2018**, *20*, 990. [CrossRef]
49. Ke, M.; Yuan, F.; Cheng, E. Underwater acoustic target recognition based on supervised feature-separation algorithm. *Sensors* **2018**, *18*, 4318. [CrossRef] [PubMed]